Recent Advances in
Modelling and Control of Stochastic Systems

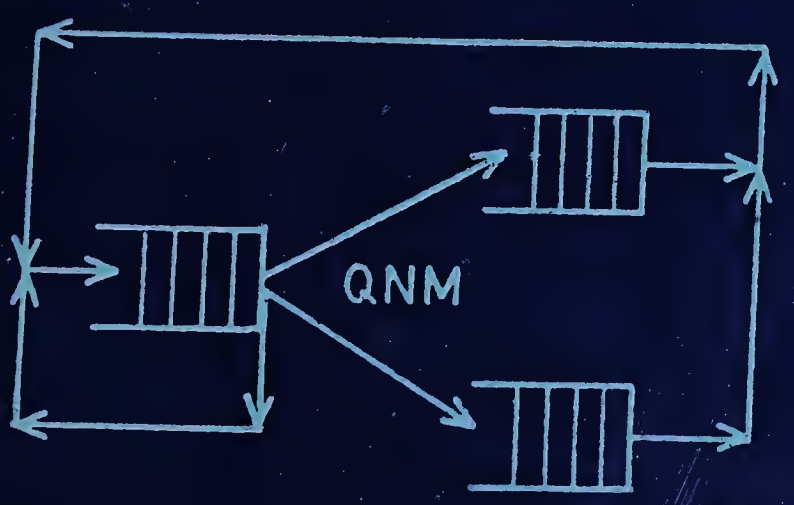
Selected Papers from a Workshop sponsored by

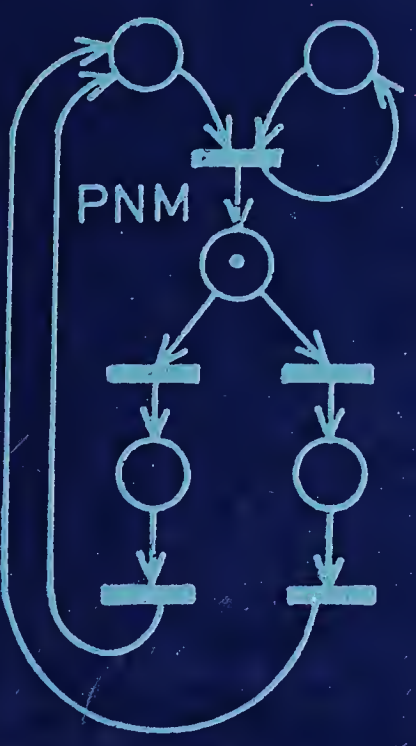
Jawaharlal Nehru Centre for Advanced Scientific Research

7 - 11 January 1991, Bangalore

Edited by

N VISWANADHAM and V S BORKAR





INDIAN ACADEMY OF SCIENCES
Bangalore 560 080

Digitized by the Internet Archive
in 2018 with funding from
Public.Resource.Org

https://archive.org/details/modellingcontrol00unse

Recent Advances in
Modelling and Control of Stochastic Systems

Selected Papers from a Workshop sponsored by
Jawaharlal Nehru Centre for Advanced Scientific Research
7 - 11 January 1991, Bangalore

Edited by
N VISWANADHAM
V S BORKAR
Indian Institute of Science, Bangalore

1991
INDIAN ACADEMY OF SCIENCES
Bangalore 560 080

The cover shows the three analytical modelling paradigms of stochastic systems, viz. Markov chains, queueing networks and stochastic Petri nets.

© 1991 by the Indian Academy of Sciences

Reprinted from *Sādhanā* – Academy Proceedings in Engineering sciences
Volume 15, pp. 249–420, 1991

Edited by V S Borkar & N Viswanadham, phototypeset at Thomson Press (India) Ltd., New Delhi and printed for the Indian Academy of Sciences at Printek Printers, Bangalore 560 079.

CONTENTS

Recent Advances in Modelling and Control of Stochastic Systems

Foreword v

Learning Automata and Neural Networks

P S SASTRY: Stochastic networks for constraint satisfaction and optimization 1

M A L THATHACHAR: Stochastic automata and learning systems 13

M VIDYASAGAR: An analysis of the equilibria of neural networks with linear interconnections 23

Stochastic Models for Manufacturing Systems

R RAM and N VISWANADHAM: Stochastic analysis of versatile workcentres 41

KARMESHU: Queueing models with threshold phenomena: With special reference to machine inteference problems 69

Y NARAHARI, N VISWANADHAM, C R MEENAKSHISUNDARAM and P HANUMANTHA RAO: Integrated analytical models for flexible manufacturing systems 81

Y NARAHARI, N VISWANADHAM and K R KRISHNA PRASAD: Markovian models for deadlock analysis in automated manufacturing systems 93

Queueing Models for Performance Analysis of Computer and Communication Systems

KIRAN M REGE: Multi-class queueing models for performance analysis of computer systems 105

VINOD SHARMA: Stability of multiple access communication networks 115

ANURAG KUMAR: Task allocation in multiserver systems – A survey of results 131

Identification and Control of Stochastic Systems

WEI REN and P R KUMAR: Recent results on least squares-based adaptive control of linear stochastic systems in white noise 147

V S BORKAR: Controlled Markov chains with constraints 155

Author Index 165

Subject Index 168

Recent Advances in modelling and Control of Stochastic Systems

Foreword

This book is a collection of papers presented at the Workshop on Recent Advances in Modelling and Control of Stochastic Systems held at the Indian Institute of Science, Bangalore, during 7–11 January 1991. The Workshop was jointly sponsored by the Jawaharlal Nehru Centre for Advanced Scientific Research, the Department of Computer Science and Automation, Indian Institute of Science, and the Central Research Laboratories of the Bharat Electronics Limited.

The prime motivation for organizing the Workshop and this special volume was a feeling that there are already enough people in India working on the frontiers of these areas, and even more people interested in them, to provide a 'critical mass' and that events like the Workshop could trigger high quality research based on exchange of ideas. The response we received at the Workshop suggests that we were not wrong on the first count. We shall feel truly rewarded for our efforts if the latter becomes a reality.

In this book we have papers on four core areas of stochastic modelling which have received considerable attention in recent years. These are:

1) Learning Automata and Neural Networks.
2) Identification and Control of Stochastic Systems.
3) Queueing Models for Performance Analysis of Computer and Communication Systems.
4) Stochastic Models for Manufacturing Systems.

There are three papers in the category 'Learning automata and neural networks'. P S Sastry presents in his paper a tutorial survey of Boltzmann machines and related stochastic network algorithms for constraint satisfaction and optimization and gives an account of his own work on the labelling problem using these techniques. Thathachar's survey article gives an extensive state-of-the-art account of learning in stochastic automata. M Vidyasagar presents his recent work on analysing equilibria of deterministic neural network models.

The topic 'Models for manufacturing systems' accounts for four articles. The article by Karmeshu applies stochastic catastrophe theory to study threshold phenomena in queueing models typically arising in machine interference problems. Ram and Viswanadham give an analysis of a versatile workcentre based on GI/G/I and multiqueue polling models with performance criteria such as mean lead time, mean inventory level and mean machine utilization. Narahari, Viswanadham and Krishna Prasad discuss the application of Markov chains with absorbing states to the modelling of deadlocks in automated manufacturing systems, primarily to determine the distribution of time to deadlock. In another paper, Narahari *et al* present a technique for combining the queueing network and Petri net models to evolve

computationally more efficient schemes for the performance analysis of flexible manufacturing systems.

There are three articles concerning 'Queueing models for performance analysis of computer and communication systems'. Kiran Rege briefly surveys techniques used for the analysis of multiclass queueing models and illustrates by a case study how a judicious mix of these can be used to advantage in practical situations. Anurag Kumar presents a state-of-the-art survey of task allocation problems in multiserver systems. Vinod Sharma gives a detailed account of recent advances in stability analysis of multiple access communication systems based on sophisticated techniques from the theory of stochastic processes.

Under 'Identification and control of stochastic systems', there are two contributions. In their article, Ren and Kumar give an account of the most recent developments in the least squares-based adaptive control of linear stochastic systems in white noise. V S Borkar reports some work on long-run average cost control of Markov chains on a countable state space when some additional cost functionals are required to satisfy prescribed bounds.

To conclude, we would like to thank the many individuals and agencies whose help and support made this publication possible. We thank Prof C N R Rao, Director, Indian Institute of Science, for his keen interest and support from the Jawaharlal Nehru Centre for Advanced Scientific Research. We are indebted to Prof R Narasimha for his help and encouragement in editing the special issues of *Sādhanā* and this book. We also would like to place on record the support we received from Mr R Ram in a variety of ways.

N VISWANADHAM
V S BORKAR
Editors

Stochastic networks for constraint satisfaction and optimization

P S SASTRY

Department of Electrical Engineering, Indian Institute of Science, Bangalore 560012, India

Abstract. Stochastic algorithms for solving constraint satisfaction problems with soft constraints that can be implemented on a parallel distributed network are discussed in a unified framework. The algorithms considered are: the Boltzmann machine, a Learning Automata network for Relaxation Labelling and a formulation of optimization problems based on Markov random field (MRF) models. It is shown that the automata network and the MRF formulation can be regarded as generalisations of the Boltzmann machine in different directions.

Keywords. Neural networks; Boltzmann machine; learning automata; consistent labelling problem.

1. Introduction

Many problems in computer vision and pattern recognition can be solved by a process that searches an appropriate space to find a point satisfying a given set of constraints. A special feature of such problems is that the constraints may be 'soft'; that is, it is not possible to categorically assert that a proposed solution satisfies or does not satisfy the constraints. The solution is more appropriately viewed as 'maximizing' the 'degree of satisfaction' of the constraints and hence such constraint satisfaction algorithms have much in common with optimization techniques. Constraint satisfaction has been used in computer vision for a long time (Waltz 1975, pp. 19–91; Davis & Rosenfeld 1981). The importance of soft constraints and the relationship with optimization have been pointed out by many people (Hummel & Zucker 1983; Blake & Zisserman 1987). Here our interest is in stochastic network-based solutions to such constraint satisfaction and optimization problems. Such networks can, by virtue of their massive parallelism, potentially deliver large processing rates needed for good performance in areas such as computer vision.

We begin our discussion with the Boltzmann machine (Hinton *et al* 1984; Rumelhart & McClelland 1986), a simple stochastic neural network that can solve some optimization problems in parallel. By incorporating simulated annealing we can reach global optima through a parallel algorithm. After this we show how this machine can be viewed as solving certain constraint satisfaction problems with soft constraints. Then we give a rigorous formulation of such problems and present a stochastic network, using learning automata, to solve them.

2. The Boltzmann machine

Boltzmann machine is a neural network whose units (or neurons) have stochastic input–output functions. Let the network have N units. The state of unit i, which is also its output, will be denoted by s_i, $1 \leqslant i \leqslant N$, and $s_i \in \{0, 1\}$, $\forall i$. The global state of the network is a vector of states of all units. Every unit in the network is connected to every other unit. Let w_{ij} denote the weight associated with the connection from the jth unit to the ith unit. The connections are symmetric; that is, $w_{ij} = w_{ji} \forall i,j$. At any instant k, the net input into the ith unit, denoted by $\text{net}_i(k)$, is calculated as

$$\text{net}_i(k) = \sum_j w_{ij} s_j(k) - \tau_i, \tag{1}$$

where τ_i is called the threshold of the ith unit.

The state of a unit depends on its net input. Units change their state asynchronously in discrete time. At any given instant of time only one randomly chosen unit changes its state. This dynamics is governed by a probabilistic law. Suppose at instant k, unit i is chosen for changing state. Then,

$$p_i = \text{Prob}\,[s_i(k+1) = 1] = 1/\{1 + \exp(-\beta \text{net}_i(k))\} \tag{2}$$

where β is a positive constant.

It is easy to see that the probability of s_i being 1 is more than 0·5 if net input is positive and less than 0·5 if net input is negative. Thus, from (1) and (2), s_i being in state 1 is more probable when the input reaching the ith unit from other units exceeds its threshold. Except for the probabilistic state transition, as given by (2), this network is very similar to the Hopfield Net (Hopfield & Tank 1985). In the Hopfield model also the net input is calculated using (1); but the state is set to 1 if net input is positive and set to zero otherwise.

Let us denote the global state of the machine at k by $S(k) = [s_1(k), \ldots, s_N(k)]^t$. We denote specific global states by S_α, S_β, etc. and $S_\alpha = [s_1^\alpha \ldots s_N^\alpha]^t$, $s_i^\alpha \in \{0, 1\}$ $\forall i$. Thus the state space of the machine is the set of all N-bit binary numbers. By (2), it is clear that $S(k)$ is a Markov chain on this state space. It is easy to show that the steady state distribution of this Markov chain is given by

$$\pi_\alpha = \exp(-\beta E_\alpha)/Z, \tag{3}$$

where

$$E_\alpha = -\tfrac{1}{2} \sum_i \sum_j w_{ij} s_i^\alpha s_j^\alpha + \sum_i s_i^\alpha \tau_i. \tag{4}$$

$\pi_\alpha$is the probability, at steady state, of finding the machine in the global state S_α. Z is a normalising constant. E_α, given by (4), is called the energy of global state S_α. From (3), it is clear that, at equilibrium, global states with lower energy are more probable.

With the energy as defined by (4), it is now possible to reinterpret the dynamics of the machine, given by (2), in an optimization context. The net input to unit i, given by (1), is $E_\alpha - E_\beta$ where the global states S_α and S_β are such that $s_i^\alpha = 0$, $s_i^\beta = 1$ and $s_j^\alpha = s_j^\beta = s_j(k) \forall j \neq i$. By (2), the ith unit prefers state 1 (i.e. $p_i > 0{\cdot}5$) if net_i is positive. Thus the ith unit prefers state 1 if its assuming state 1 will decrease the overall energy. It may be noted that (2) allows a state change by an ith unit that increases the overall

energy also, though with a smaller probability. Contrasting this with the Hopfield model, we see that all state changes allowed there have to decrease the global energy. This is true of all gradient descent algorithms that continuously try to decrease the objective function. But in the Boltzmann machine we allow 'uphill moves' also; that is, we allow state changes that increase the energy (or equivalently the value of the objective function). Once again referring to (2), we see that the probability with which uphill moves are allowed depends on the value of the parameter β.

3. Boltzmann machine as an optimization technique

We are now in a position to appreciate the optimization capabilities of the Boltzmann machine. Suppose we have to determine values for N Boolean variables so as to minimize a quadratic objective function. Then we can view the given objective function as our energy and, by putting it in the form of (4), we can decide on the weights w_{ij}. If we now run a Boltzmann machine with these weights, then, at equilibrium, the machine will spend more time in states that result in lower values of the objective function. So, we can pick up minima by, for example, gathering statistics about state occupancy (see, for example, Marroquin *et al* 1987, for an application of this concept in computer vision).

As an example, let us consider the Travelling Salesman Problem (TSP). Given a set of cities and distances between pairs of them, the problem is to find a tour, i.e., a sequence of visiting all the cities without visiting any city twice, so that the tour length, i.e., total distance travelled, is minimum. We can think of any tour as an assignment of a position number to each city. If we have N cities and have one unit to represent each city then we need units that are capable of being in N distinct states so that the global states can be viewed as a possible tour. But since our units take only 0–1 values, we have to reformulate the problem involving only binary decision variables. We will choose our variables as s_{xi}, $1 \leqslant x, i \leqslant N$. The variable s_{xi} governs the decisions of putting city x in position i. We denote by $w_{xi,yj}$, the weight connecting the units xi and yj. It is easy to see that not every possible global state represents a tour. For example, if for a given x, $s_{xi} = 1$ for more than one i then it means we are assigning more than one position to a given city and that is not permitted. Hence in formulating our energy, in addition to incorporating the tour length, we should also incorporate constraints which make sure that energy is large for global states that do not represent a valid tour. With this motivation, a possible energy function is

$$E(S) = (A/2)\sum_x \sum_i \sum_{j \neq i} s_{xi}s_{xj} + (B/2)\sum_i \sum_x \sum_{y \neq x} s_{xi}s_{yi} + (C/2)\left(\sum_x \sum_i s_{xi} - N\right)^2 +$$

$$+ (D/2)\sum_x \sum_{y \neq x} \sum_i d_{xy}s_{xi}(s_{y,i+1} + s_{y,i-1}) \qquad (5)$$

The first term penalises (i.e. results in higher energy of) any global state where more than one position is assigned to the same city. The second term similarly penalises global states that assign two different cities to the same position. Now, if we take $s_{xi} = 0 \ \forall \ x, i$, then the first two terms are zero but we still have not got a tour. So, the third term enforces the condition that there should be exactly N cities on the tour. These three terms together incorporate the conditions for valid tours. The fourth term accounts for the actual length of the tour represented by a global state. Here

d_{xy} is the distance between cities x and y, and the subscripts $i+1$ and $i-1$ should be understood such that $N+1$ is 1 and $1-1$ is N.

If we rewrite (5) in the form of (4) (where w_{ij} now becomes $w_{xi,yj}$), we get

$$w_{xi,yj} = -A\delta_{xy}(1-\delta_{ij}) - B\delta_{ij}(1-\delta_{xy}) - C - Dd_{xy}(\delta_{j,i+1} + \delta_{j,i-1}), \qquad (6)$$

and

$$\tau_{xi} = CN,$$

where

$$\delta_{ij} = 1, \text{ if } i = j,$$
$$= 0, \text{ otherwise.}$$

By proper choice of parameters A, B, C, D one can make sure that global states corresponding to a tour have lower energy than others. Further, due to the last term in (5), tours with smaller lengths have lower energy. If we run the Boltzmann machine with weights and thresholds as given by (6), then, at steady state, the machine will spend more time in global states that correspond to tours with lower length. If we gather statistics about state occupancy at steady state, we can find the solution to the TSP by picking the state with highest probability.

The procedure as outlined above, though offering a solution to an optimization problem in principle, is not very effective in many cases. In TSP, for example, there are 2^{N*N} possible global states and all of them have non-zero steady state probabilities [see (3)]. Hence the probability of finding the machine in the optimal tour might be numerically very small (though still greater than finding the machine in any other state). Thus one needs statistics about a larger number of states gathered over a long time to be able to obtain the solution.

A possible method to overcome at least the storage overheads of gathering statistics, is the use of simulated annealing (Kirkpatrick *et al* 1983).

As explained earlier, the dynamics of the Boltzmann machine allows state changes that increase the energy. The probability with which these uphill moves are made is controlled by the parameter β in (2). If $\beta = 0$ then all state changes (both 'uphill' and 'downhill') are equally probable. On the other hand, as β tends to infinity only those state changes that reduce the energy are possible. In the terminology of simulated annealing, $1/\beta$ is called temperature. To incorporate simulated annealing into the Boltzmann machine, we start the machine with a high temperature (low value of β) where state changes are essentially random. Then slowly, the temperature is decreased with time (i.e. β is increased with k) ultimately approaching zero temperature (β tending to infinity). This process is called cooling. If the cooling schedule, that is, the manner of decreasing temperature with time, is sufficiently slow then, at equilibrium, the machine will be in only those global states which represent global minima of the energy function. A cooling schedule that is sufficient is $T(k) \geqslant A/(B + \log k)$ where A and B are appropriate constants (Mitra *et al* 1986). Here $T(k)(= 1/\beta(k))$ is the temperature at instant k. It is easy to see that the temperature needs to be reduced very slowly and hence a large number of iterations are needed to get the temperature sufficiently near zero.

Thus the Boltzmann machine with simulated annealing can find the global minima of quadratic objective functions over Boolean variables but will take enormous time. Unlike the case where the machine is run at constant temperature (and hence the need to gather statistics at the steady state), here the space complexity of the optimization algorithm is very small.

In conclusion, we can see that the Boltzmann machine is an interesting network-based optimization technique. It may be noted that by making the net input of a unit equal to the energy difference between appropriate global states (as explained in § 2), we can think of this as a general technique of optimization. But only in the case of quadratic functions can we write this energy difference as in (1). Viewing this computation as being performed on a network, where each unit decides its state based on the weighted sum of outputs of other units, is possible only when the energy function is quadratic. It may appear that there is not much parallelism here because at each instant only one unit is updated. But if we let each unit be run with its own clock, whose ticks are Poisson-distributed and which is independent of all other clocks, then we can implement the network on a set of processors without need for any explicit synchronism. To that extent it can be viewed as a parallel network.

4. Boltzmann machine and soft constraints

Consider a Boltzmann machine where the states s_i are such that $s_i \in \{-1, 1\}\ \forall i$. Now if we consider the global states S_α and S_β such that $s_i^\alpha = -1$, $s_i^\beta = 1$ and $s_j^\alpha = s_j^\beta\ \forall j \neq i$, then, forgetting the thresholds for the time being, we get $E_\alpha - E_\beta = 2\Sigma_j w_{ij} s_j$. Thus even now, the net input measures the change in global energy and hence by the dynamics given by (2), the machine still prefers downhill moves in trying to minimise the energy given by (4).

Now, the energy to be minimised is $-\Sigma w_{ij} s_i s_j$. Therefore, if w_{ij} is negative it is worthwhile making s_i ad s_j of opposite sign and vice-versa. Thus w_{ij} can be thought of as a constraint on the possible values for s_i and s_j. But this constraint is soft. Thus, just because w_{ij} is negative it may not always be desirable to make s_i and s_j of opposite sign. For example we can have weights w_{ki} and w_{kj} both of which are negative and much larger than w_{ij}. Hence, it is desirable to make the pairs s_i and s_k and s_j and s_k of opposite sign which forces us to make s_i and s_j of the same sign though w_{ij} is negative.

In the next section we give a formulation of such constraint satisfaction problems. If the objective is only to find global minimum of the energy, then the view of constraints satisfaction will not be very appealing. (A constraint satisfaction problem is appealing mostly in the context where there are multiple solutions that 'satisfy' the constraints.) So, we establish a connection between constraint satisfaction and local optima of the energy. Since the energy is defined on a finite set we have to clearly define what we mean by local optima. All these will be discussed in detail in the next section.

5. A mathematical formulation of constraint satisfaction

In formulating problems with soft constraints we follow the treatment of Hummel & Zucker (1983). This class of problems is termed the labelling problems. A general labelling problem is specified by giving

i) A set of objects, $\mathbf{O} = \{O_1, \ldots, O_N\}$
ii) A set of labels, $\Lambda = \{1, \ldots, M\}$
iii) A neighbour relation over $\mathbf{O}$ specifying which pair of objects constrain each other.
iv) A set of compatibility functions that specify the constraints.

The problem is to assign a label to each object such that the assignment is 'consistent' with respect to the compatibility functions. For every pair of objects O_i and O_j that constrain each other (as specified by the neighbour relation), there is a compatibility function, $r_{ij}: \Lambda \times \Lambda \to R$. $r_{ij}(\lambda, \lambda')$ is a measure of consistency of object-label pairs (O_i, λ) with (O_j, λ'). Thus this compatibility specifies, locally, how well the label λ on O_i fits the decision of putting the label λ' on O_j. We can assume that r_{ij} are defined for all pairs i and j and stipulate, as a notation, that $r_{ij} \equiv 0$ if O_i and O_j are not neighbours.

Once such local compatibility functions are given, we have to define what we mean by global consistency. Let us denote by $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)$, a label assignment that assigns label λ_i to object O_i, $1 \leqslant i \leqslant N$.

DEFINITION 1

A label assignment $(\lambda_1, \ldots, \lambda_N)$ is said to be consistent if

$$\sum_j r_{ij}(\lambda_i, \lambda_j) \geqslant \sum_j r_{ij}(\lambda, \lambda_j), \forall \lambda, i = 1, \ldots, N.$$

It is said to be strictly consistent if the above inequalities are strict for $\lambda \neq \lambda_i$.

We can think of the quantity $\Sigma r_{ij}(\lambda_i, \lambda_j)$ as the amount of evidence in support of label λ_i on O_i given the labels on all other objects. Thus at a consistent labelling if we change the label on any one object, the 'amount of consistency' decreases. That is, changing the label on O_i will decrease net supporting evidence at O_i and this is true for all objects. This can also be viewed as a local maximum of an appropriate objective function.

Let L denote the space of all labellings. L will be an N-fold Cartesian product of the label set, Λ. We define a neighbourhood structure on L as follows.

DEFINITION 2

Let $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N) \in L$ and let us denote by $N(\boldsymbol{\lambda})$ all neighbours of $\boldsymbol{\lambda}$. Then for any $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_N) \in L$, $\boldsymbol{\mu} \in N(\boldsymbol{\lambda})$ if and only if there exists i, $1 \leqslant i \leqslant N$, such that $\lambda_i \neq \mu_i$ and $\lambda_j = \mu_j \; \forall j \neq i$.

Thus given a labelling $\boldsymbol{\lambda}$, any other labelling differing from it in the assignment of a label to only one object, is a neighbour of $\boldsymbol{\lambda}$. Now we define a functional, F, on L by

$$F(\boldsymbol{\lambda}) = \sum_{i,j} r_{ij}(\lambda_i, \lambda_j). \tag{7}$$

DEFINITION 3

A labelling $\boldsymbol{\lambda} \in L$ is said to be a local maximum of F if

$$F(\boldsymbol{\lambda}) \geqslant F(\boldsymbol{\mu}), \text{ for all } \boldsymbol{\mu} \in N(\boldsymbol{\lambda}).$$

Now we are ready to establish the connection between consistency as defined by definition 1 and the local maxima of F.

Theorem 1. *Let the compatibility functions be symmetric. That is,* $r_{ij}(\lambda, \lambda') = r_{ji}(\lambda', \lambda)$, *for all* i, j, λ, λ'. *Then, a labelling* $\boldsymbol{\lambda} \in L$ *is consistent* (*definition* 1) *if and only if it is a local maximum of the functional* F (*definition* 3).

The proof of this theorem is very straightforward and directly follows from the

definitions. Now let us take another look at the Boltzmann machine to understand it in the labelling framework.

Consider a labelling problem with N objects and the label set $\{+1, -1\}$. Let the compatibility functions be $r_{ij}(s_i, s_j) = w_{ij}s_i s_j$. We now apply definition 1 to determine when a labelling (global state), S, is consistent. We must have

$$\sum_j r_{ij}(s_i, s_j) \geqslant \sum_j r_{ij}(s, s_j).$$

Since each s is ± 1, substituting the values for r_{ij} we get, for the case when $s_i = +1$,

$$\sum w_{ij}s_j > -\sum w_{ij}s_j (s = -1 \neq s_i),$$

i.e.

$$\sum w_{ij}s_j > 0.$$

Similarly we get, if $s_i = -1$,

$$-\sum w_{ij}s_i > \sum w_{ij}s_j,$$

i.e.

$$\sum w_{ij}s_j < 0.$$

Hence, for a global state S to be consistent, s_i should be of the same sign as $\Sigma w_{ij}s_j$. That is, we want to make $s_i > 0$ if the net input is positive and vice-versa. That is exactly the same as the Hopfield net. In the Boltzmann machine we said that we do not make $s_i > 0$ with probability 1 every time the net input is greater than zero because we want to include uphill moves also to avoid local minima. To understand this let us look at the F function for this case,

$$F(S) = \sum_{i,j} w_{ij}s_i s_j.$$

Thus $F(.) = -E$. We know that the consistent labellings are local maxima of F (in the sense of definition 3). Thus they are local minima of E. Hence the consistency as given by definition 1 makes sure that we reach a local minimum of the energy function E. However, by allowing uphill moves and using simulated annealing, the Boltzmann machine can, in principle, find the global minima of energy E.

The labelling problem as formulated is an extension to the set of problems that can be tackled by the Boltzmann machine. Here we allow an arbitrary number of labels rather than restrict each object to a binary state. Therefore, even though consistency guarantees only a local maximum of F, due to the extra freedom allowed in the formulation, it might often be a more preferable alternative to formulating the problem in the framework of a Boltzmann machine. For example, in the case of TSP we have seen that it is essentially the need to work with Boolean variables that force us to add many constraints as penalty terms in the energy. By choosing the set of objects and labels intelligently we may avoid many such penalty terms in our energy function and thus even a local maximum of F might be an acceptable solution. The second reason for investigating labelling problems is that we can get parallel network-based algorithms to solve them so that the time complexity is much smaller than that of the Boltzmann machine.

6. A network of learning automata for solving the labelling problem

In this section we present a stochastic network based on learning automata for solving the consistent labelling problem. A learning automaton is an adaptive decision-making device that can identify the optimal action out of a finite set of actions through interactions with a random environment. We will not be giving any details regarding general learning automata models and the reader is referred to Narendra & Thathachar (1989) for such details. The automata algorithm for the labelling problem is presented in Thathachar & Sastry (1986). Here our interest is in viewing it as a stochastic network and contrasting it with other similar stochastic networks.

Consider a labelling problem with N objects and M labels. We use a network of N automata to solve the problem. The state of each automaton is a probability distribution over the set of labels. The state of the automaton i (associated with object O_i) at instant k is $\mathbf{p}_i(k) = [p_{i1}(k) \ldots p_{iM}(k)]^t$ where p_{is} is the probability with which automaton i chooses (i.e. associates object O_i with) label s. The output of the automaton is a random realisation of this probability distribution. That is, the output of automaton i at instant k is a choice of label for object O_i, at random, based on $\mathbf{p}_i(k)$. Each automaton receives at its input the outputs of other automata (i.e. labels chosen for other objects) and uses this information to update its state, i.e., its label probability distribution. The network functions synchronously. That is, at each instant all the automata, simultaneously and independently, select labels for their objects at random based on their label probability distributions. Then each of them receives at its input the outputs of other automata, computes their net input and uses the net input to update the states. The state updating is also done simultaneously by all the automata. We now specify the dynamics of the network, that is, rules for computing net input and updating the state.

Suppose at instant k, the ith automaton has chosen label λ_i. Then the net input to automaton i at k, $\text{net}_i(k)$, is given by

$$\text{net}_i(k) = (1/N) \sum_j r_{ij}(\lambda_i, \lambda_j). \tag{8}$$

The state of the automaton at $(k+1)$ is computed as

$$\begin{aligned} p_{i\lambda_i}(k+1) &= p_{i\lambda_i}(k) + a\,\text{net}_i(k)(1 - p_{i\lambda_i}(k)), \\ p_{ij}(k+1) &= p_{ij}(k) - a\,\text{net}_i(k) p_{ij}(k), \quad j \neq \lambda_i, \end{aligned} \tag{9}$$

where $0 < a < 1$ is a constant.

The net input into an automaton is still given by a summation of the effects of outputs of other automata. But instead of a simple weighted sum, we use a 'nonlinear' summation using the functions r_{ij}.

In the terminology of Learning Automata theory, net_i is called the reaction received by the ith automaton from its environment and the state updating given by (9) is called Linear Reward Inaction algorithm. Since $\mathbf{p}_i(k+1)$ should be a probability vector, it is necessary that $0 \leqslant \text{net}_i(k) \leqslant 1$ for all i and k. To ensure this we assume, without loss of generality (Thathachar & Sastry 1986) that $r_{ij}(\lambda, \lambda') \in [0, 1]$ for all i, j, λ, λ'.

Let us denote by $P(k) \in R^{MN}$, the collection of all label probability vectors $\mathbf{p}_i(k)$,

$i = 1, \ldots, N$. Since each $\mathbf{p}_i$ is a probability vector, $P(k)$ belongs to $K \subset R^{MN}$, defined by

$$K = \{P \in R^{MN} : P = (p_1^t \ldots p_N^t)^t, p_i = (p_{i1} \ldots p_{iN})^t \in R^M,$$
$$p_{ij} \geqslant 0, \forall i,j, \sum_j p_{ij} = 1, i = 1, \ldots, N\}.$$

We can think of K as the N-fold cartesian product of M-dimensional simplices. Let $\mathbf{e}_i$ denote an M-dimensional unit (row) vector with its ith component unity. $(\mathbf{e}_{\lambda_1} \ldots \mathbf{e}_{\lambda_N})^t \in K$ is, for our algorithm, a labelling $(\lambda_1, \ldots, \lambda_N)$ that assigns the label λ_i to object i, $i = 1, \ldots, N$. All such points are termed corners of the space K.

$P(k)$ is the global state of our network at instant k. By the dynamics specified by (8) and (9), $P(k)$ is a Markov process with state space K. To understand the performance of our network we need to obtain the asymptotic behaviour of $P(k)$. For this, consider a piecewise-constant continuous time interpolation of $P(k)$, $\tilde{P}^a(.)$, defined by

$$\tilde{P}^a(t) = P(k) \text{ for } t \in [ka, (k+1)a),$$

where a is the parameter used in (9).

$\tilde{P}^a(\cdot) \in D^{MN}$, the space of all functions from R into R^{MN} which are left-continuous and have right-hand limits. Under the Skorohod metric (Billingsley 1968) this is a complete metric space. Now consider the sequence $\{\tilde{P}^a(\cdot)\}$ indexed by a. Using weak convergence results, it can be proved (Kushner 1984; Thathachar & Sastry 1986) that $\tilde{P}^a$ converges weakly to $\tilde{P}^0$, where $\tilde{P}^0$ is a solution of the ordinary differential equation (ODE)

$$z = g(z), \quad z(0) = P(0), \tag{10}$$

where

$$ag(z) = E[P(k+1) - P(k) | P(k) = z].$$

By analysing the asymptotic behaviour of the ODE (10), and by the properties of weak convergence, one can obtain the asymptotic properties of $P(k)$. The is stated in the following theorem.

Theorem 2. *If the value of the parameter a in* (9) *is sufficiently small then the following is true about the asymptotic behaviour of the dynamical process* $\{P(k), k \geqslant 1\}$.

(i) *Every strictly consistent labelling is an asymptotically stable stationary point of the process.*
(ii) *Every stable corner of the state space K is a consistent labelling.*
(iii) *All stationary points in the interior of K are unstable.*

The implication of this theorem is that the network mostly converges to a corner in K that is a consistent labelling. The convergence result is incomplete because we do not know about the stability or otherwise of stationary points on the faces of K (i.e., points that are neither corners nor are in the interior). Also, the ODE being nonlinear, there is a possibility of other limiting behaviour such as limit cycles. In practice, these cases never seem to arise.

6.1 *Comparison with the Boltzmann machine*

In the Boltzmann machine the global state is an ergodic Markov chain. Thus, at any finite temperature, the machine converges to a probability distribution over the set

of all labellings. The global state of the automata network is a Markov process with absorbing states. [It can easily be verified that all corners of K are absorbing for $P(k)$.] Thus the automata network converges to a specific labelling which is one of the possibly many consistent labellings.

The steady state probability distribution of the Boltzmann machine prefers low energy states. Though the machine keeps visiting all labellings with a non-zero probability, the probability of visiting states with lower energy is higher. By incorporating simulated annealing we can reach the global minimum; but it will take a very long time. In constrast, the automata network converges to a specific labelling that is a local maximum of $F(.)$ This may or may not a global maximum depending on $P(0)$. But there is no scope for incorporating a mechanism like simulated annealing into the automata network.

For the Boltzmann machine, viewed as an optimizer, the objective function has to be quadratic involving only binary variables. In the more general labelling problem, the labels need not be binary, and also, by clever choice of r_{ij}'s one can tackle nonquadratic functions as well. By utilising the extra freedom given by the labelling framework one can reduce the possible unwanted local minima as compared to the energy function formulation of the Boltzmann machine. For example, in the energy function given by (5), we had three terms for incorporating the constraints which are necessary when formulating TSP using only Boolean variables. In a labelling framework, suppose we let each automaton represent a city and the actions represent position numbers. Then we do not need constraints such as no city be placed in more than one position (the 'A-term' in (5)) and every city be placed in some position (the 'C-term' in (5)). This can effectively be used in reducing the unwanted local minima (Muralidharan 1989).

As mentioned earlier, the automata network is primarily for constraint satisfaction problems with soft constraints. In such problems there will be, in general, many consistent solutions and what is desired is a consistent labelling 'close' to the initial problem data provided. Many problems in computer vision are of this type. In the automata algorithm the initial data can be incorporated through $\mathbf{p}_i(0)$, the initial values for the action probabilities thereby biasing the network to converge to a consistent labelling closeby. Since the Boltzmann machine results in an ergodic chain, the initial state has no effect on the asymptotic behaviour and hence we have to code the initial data also as part of the energy function.

This algorithm was successfully employed in computer vision problems such as stereopsis and object recognition (Sastri *et al* 1988). Also unlike the Boltzmann machine, where only one unit changes state at any given instant, here all automata function synchronously and hence the network can be implemented in parallel, for example, on an single instruction multiple data (SIMD) machine (Banerjee *et al* 1987).

6.2 *Another stochastic network for the labelling problem*

An important and widely studied algorithm for the constraint satisfaction and optimization problems of the kind studied in this paper is the algorithm of Geman & Geman (1984).

Consider a labelling problem as specified at the beginning of § 5. The neighbour relationship can be viewed as a graph with the set of objects as vertices. A clique of this graph is a set of objects, where all pairs of objects are neighbours. Instead of

specifying a set of compatibility functions r_{ij}, let us assume that the prior knowledge about the constraints to be satisfied by the solution can be encoded through an energy function defined on the set of labellings by

$$E(\lambda) = \sum_{c \in \mathscr{C}} V_c(\lambda), \tag{11}$$

where $\mathscr{C}$ is the set of all cliques. $V_c(\boldsymbol{\lambda})$ is called a potential function and its value depends only on the labels (components of $\boldsymbol{\lambda}$) of those objects which are contained in the clique c. If we allow only cliques containing pairs of objects then V_c is similar to $r_{ij}(\lambda_i, \lambda_j)$ and $E(.)$ is similar to $F(.)$ defined by (7). So the local minima of $E(.)$ can be thought of as the 'consistent' labellings. We define a prior probability for each labelling by

$$\pi(\boldsymbol{\lambda}) = \exp(-E(\boldsymbol{\lambda})/Z, \tag{12}$$

where Z is a normalising constant. It may be noted that this is the same as the steady state distribution of a Boltzmann machine though here the labels are not binary. The prior probability distribution given by (12) prefers labellings with low energy. As earlier, the objective is to find a consistent labelling that is close to the initial data given. In this approach we define the solution to be the global maximum of posterior probability $P(\boldsymbol{\lambda}|\mathbf{d})$, which is the probability that the 'correct' labelling is $\boldsymbol{\lambda}$ given the data $\mathbf{d}$. We can use Bayes rule to calculate $P(\boldsymbol{\lambda}|\mathbf{d})$ using (12) if we have the conditional probabilities, $P(\mathbf{d}|\boldsymbol{\lambda})$. These can be specified by having a model for the process which generates the data. Under some mild conditions on the conditional probabilities (which are generally true for image processing problems), $P(\boldsymbol{\lambda}|\mathbf{d})$ will have the same form as (12) with the posterior energy containing one term for the data in addition to the summation of clique potentials as in (11). So now the problem is to find a parallel algorithm for finding the global maximum of a probability distribution as in the case of the Boltzmann machine. In the algorithm we have a unit for each object which chooses a label at random using the conditional probabilities given the current labels of all the neighbouring objects. This so-called Gibbs sampler algorithm can be shown to generate an ergodic Markov chain whose steady state distribution is the needed posterior probability distribution (same form as (12) but with E replaced by posterior energy). So as in the case of the Boltzmann machine we can reach the global maximum of the posterior probability by incorporating simulated annealing.

Thus the Geman & Geman (1984) algorithm can be viewed as an extension of the Boltzmann machine to the more general labelling framework. It generates an ergodic Markov chain (and hence we need to incorporate the data into the energy function) and we aim to reach the global maximum by incorporating simulated annealing. The main problem with this algorithm is its extremely slow rate of convergence.

This algorithm is presented rather briefly because it is best explained in the context of image processing problems and such a discussion is beyond the scope of this paper. The details can be found in Geman & Geman (1984), Geman & Graffigne (1987) and Geman *et al* (1990). An extensive comparison of the automata model with the Geman & Geman algorithm is found in Banerjee (1989).

Though in this paper we have discussed the automata model and the Geman & Geman algorithm as if they are extensions of the Boltzmann machine, all the algorithms have been originally proposed more or less independently from different perspectives.

7. Conclusions

In this article we have presented a brief account of some stochastic network models for solving special types of constraint satisfaction and optimization problems. The use of such soft constraints is important in areas such as computer vision, image processing etc. Also these models give a flavour of the type of problems that can be tackled by artificial neural networks. There are deterministic schemes also for tackling soft constraints, for example, the graduated non-convexity algorithm of Blake & Zisserman (1987). See Blake (1989) for a comparison of the deterministic scheme against those that use simulated annealing.

The Boltzmann machine discussed here also has applications in pattern recognition and learning. These are not discussed because it would be outside the scope of the topic of this paper. The reader can refer to Rumelhart & McClelland (1986) for details.

References

Banerjee S 1989 *On stochastic relaxation paradigms for computational vision*, Ph D thesis, Dept. Elec. Eng., Indian Institute of Science, Bangalore

Banerjee K, Sastry P S, Ramakrishnan K R, Venkatesh Y V 1988 An SIMD machine for low-level vision. *Inf. Sci.* 44: 19–50

Billingsley P 1968 *Convergence of probability measures* (New York: John Wiley)

Blake A 1989 Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-11: 2–12

Blake A, Zisserman A 1987 *Visual reconstruction* (Cambridge: MIT Press)

Davis L, Rosenfeld A 1981 Cooperating processes in low level vision. *Artif. Intell.* 17: 245–263

Geman S, Geman D 1984 Stochastic relaxation, Gibbs distribution and Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-6: 721–741

Geman D, Geman S, Graffigne C, Dong P 1990 Boundary detection by constrained optimization. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-12: 609–628

Geman S, Graffigne C 1987 Markov random field image models and their applications to computer vision. *Proc. Intl. Congress of mathematicians* (ed.) A M Gleason (Providence: Am. Math. Soc.)

Hinton G E, Sejnowsky T J, Ackley D H 1984 Boltzmann machines: Constraint satisfaction networks that learn, Tech. Rep. CMU-CS- 84–119, Carnegie Mellon Univ. Pittsburgh

Hopfield J J, Tank D W 1985 Neural computation of decisions in optimization problems. *Biol. Cybernet.* 52: 141–152

Hummel R A, Zucker S W 1983 On the foundations of relaxation labelling processes. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-5: 267–286

Kirkpatrick S, Gelatt C D, Vecchi M P 1983 Optimization by simulated annealing. *Science* 220: 671–680

Kushner H 1984 *Approximation and weak convergence methods for random processes* (Cambridge: MIT Press)

Marroquin J, Mitter S, Poggio T 1987 Probabilistic solution of ill-posed problems in computational vision. *J. Am. Stat. Assoc.* 82 (397): 76–89

Mitra D, Fabio R, Vincentelli A S 1986 Convergence and finite-time behaviour of simulated annealing. *Adv. Appl. Probab.* 18: 747–771

Muralidharan S 1989 *Parallel distributed processing schemes for combinatorial optimisation*, M Sc (Eng.) thesis, Dept. Elect. Eng., Indian Institute of Science, Bangalore

Narendra K S, Thathachar M A L 1989 *Learning automata: An introduction* (Englewood Cliffs, NJ: Prentice-Hall)

Rumelhart D E, McClelland J L 1986 *Parellel distributed processing* (Cambridge: MIT Press) Vols. 1 & 2

Sastry P S, Ramakrishnan K R, Banerjee S 1988 A local cooperative processing model for low level vision. *Indo-US Workshop on Systems and Signal Processing* (New Delhi: Oxford and IBH)

Thathachar M A L, Sastry P S 1986 Relaxation labelling with Learning Automata. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-8: 256–268

Waltz L D 1975 Understanding of line drawings of scenes with shadows. In: *Psychology of computer vision* (ed.) P H Winston (New York: McGraw-Hill)

Stochastic automata and learning systems

M A L THATHACHAR

Department of Electrical Engineering, Indian Institute of Science, Bangalore 560 012, India

Abstract. We consider stochastic automata models of learning systems in this article. Such learning automata select the best action out of a finite number of actions by repeated interaction with the unknown random environment in which they operate. The selection of an action at each instant is done on the basis of a probability distribution which is updated according to a learning algorithm. Convergence theorems for the learning algorithms are available. Moreover the automata can be arranged in the form of teams and hierarchies to handle complex learning problems such as pattern recognition. These interconnections of learning automata could be regarded as artificial neural networks.

Keywords. Stochastic automata; learning systems; artificial neural networks.

1. Introduction

Stochastic automata operating in unknown random environments form general models for learning systems. Such models are based on some early studies made by American psychologists (Bush & Mosteller 1958) and also on automata studies introduced into engineering literature by Soviet researchers (Tsetlin 1973). Stochastic automata interact with the environment, gain more information about it and improve their own performance in some specified sense. In this context, they are referred to as learning automata (Narendra & Thathachar 1989).

The need for learning exists in identification or control problems when high levels of uncertainty are present. For instance, the pole balancing problem can be regarded as a deterministic control problem when the relevant parameters of the system such as the mass of the cart and mass and length of the pole are given. When some of the parameters are unknown, it can be treated as an adaptive control problem. When even the dynamic description of the system is not used to determine the control, learning becomes necessary. In the latter approach, the input to the system is generated by experience through a learning algorithm which builds up associations between the input and output.

Learning is very much relevant in tasks such as pattern recognition where detailed mathematical descriptions are usually not available and feature measurements are noisy. Similar is the situation with regard to computer vision problems such as stereopsis.

In recent years there has been a great deal of interest in artificial neural networks. The main aim here is to use a dense interconnection of simple elements to build systems which provide good performance in perceptual tasks. The learning automata described here appear to be well-suited as building blocks of stochastic neural networks as they can operate in highly noisy environments.

2. The random environment

The learning model that we consider involves the determination of the optimal action out of a finite set of allowable actions. These actions are performed on a random environment. The environment responds to the input action by an output which is probabilistically related to the input action.

Mathematically the random environment can be defined by a triple $\{\alpha, d, \beta\}$ where

$\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ represents the input set,

$d = \{d_1, d_2, \ldots, d_r\}$, the set of reward probabilities, and

$\beta = \{0, 1\}$, a binary set of outputs.

The input $\alpha(n)$ to the environment is applied at discrete time $n = 0, 1, 2, \ldots$ etc. and belongs to the set α. The output $\beta(n)$ is instantaneously related to the input $\alpha(n)$ and is binary in the simplest case. An output $\beta(n) = 1$ is called a reward or favourable response and $\beta(n) = 0$ is called a penalty or unfavourable response. The reward probabilities relate $\alpha(n)$ and $\beta(n)$ as follows,

$$d_i = P\{\beta(n) = 1 | \alpha(n) = \alpha_i\}, \quad (i = 1, \ldots, r). \tag{1}$$

Consequently d_i represents the probability that the application of an action α_i results in a reward output. An equivalent set of penalty probabilities $c_i = 1 - d_i$ can also be defined.

$$c_i = P\{\beta(n) = 0 | \alpha(n) = \alpha_i\}, \quad (i = 1, \ldots, r). \tag{2}$$

Models in which the output of the environment is binary are called *P*-models. A generalization of this model is the *Q*-model where the output belongs to a finite set in the interval $[0, 1]$. A further generalization is the *S*-model where the output is a continuous random variable which assumes values in $[0, 1]$. *Q* and *S* models provide improved discrimination of the nature of response of the environment. The penalty probabilities are usually regarded as constants. In such a case the environment is called stationary. When c_i vary with time n directly or indirectly, the environment is nonstationary. Many practical problems involve nonstationary environments.

The problem of learning in an unknown random environment involves the determination of the optimal action from the input-output behaviour of the environment. The optimal action is the action most effective in producing the reward output or the favourable response.

Let

$$d_m = \max_i \{d_i\} \tag{3}$$

Then α_m is the optimal action and could be easily identified if each d_i is known. Hence

a meaningful learning problem exists only when the reward probabilities are unknown. In some problems the set of reward probabilities $\{d_i\}$ may be known but the action with which each reward probability is associated may not be identified. Learning the optimal action is a relevant problem even here.

Learning involves experimentation on the environment by choosing input actions and correlating them with outputs in developing a strategy for picking new actions. A stochastic automaton is very helpful in carrying out such operations in an effective manner.

3. The learning model

The stochastic automata that we consider in the learning model can be represented by the triple $\{\alpha, \beta, A\}$. $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is the finite set of outputs or actions; $\beta = \{0, 1\}$ is the binary set of inputs; A is the algorithm for updating the action probability vector $p(n)$.

$$p(n) = [p_1(n), p_2(n), \ldots, p_r(n)]^T, \tag{4}$$

where

$$p_i(n) = P\{\alpha(n) = \alpha_i\},$$

and T denotes transpose. Thus the stochastic automaton selects an output action $\alpha(n)$ at time n randomly based on $p(n)$. It updates $p(n)$ to $p(n+1)$ according to the algorithm A and selects a new action $\alpha(n+1)$.

The learning model that we study consists of the stochastic automaton in a feedback connection with the random environment (figure 1). The action $\alpha(n)$ of the automaton forms the input to the environment and the output $\beta(n)$ of the environment is the input to the automaton. An automaton acting in an unknown random environment so as to improve its performance in some specified sense is called a learning automaton.

This learning model is closely related to the two-armed bandit problem extensively studied in statistics. The two-armed bandit is a machine with two arms. A human test subject is asked to pull one of the 2 arms. The subject is rewarded or punished according to a previously determined schedule with a fixed probability of reward for the two choices. The problem is in finding whether the subject will learn to select the better arm associated with the higher probability of reward in successive experiments.

The above problem illustrates the classical dilemma encountered between identification and control in such learning situations. The subject must decide which of the 2 arms must be chosen on the basis of the previous performance. The dilemma is whether the subject should choose the arm which is known to be better so far or whether he should select the arm about which least knowledge exists so that new knowledge of relative effectiveness of the arms is obtained.

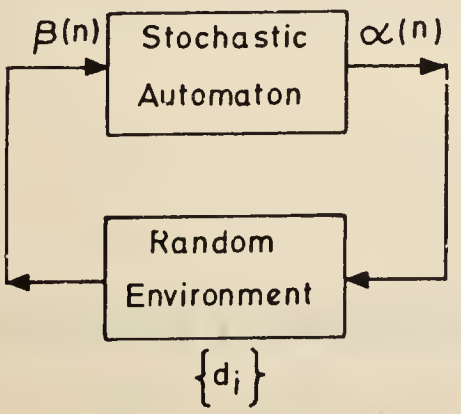


Figure 1. Stochastic automaton in random environment.

4. Norms for learning

Although learning is understood in a qualitative way, it is necessary to set up quantitative norms for a proper understanding of the model. On close scrutiny it is found that several kinds of learning can be defined. Some prominent definitions are given below.

At the outset it may be noted that the action probability vector $p(n)$ is a random variable and the sequence $\{p(n)\}$ is a random process. A quantity of much importance is the Average Reward $W(n)$ defined as

$$\begin{aligned} W(n) &= E[\beta(n)|p(n)] \\ &= \sum_{i=1}^{r} P[\beta(n)=1|\alpha(n)=\alpha_i]\, P[\alpha(n)=\alpha_i] \\ &= \sum_i d_i p_i(n). \end{aligned} \tag{5}$$

If actions are selected purely randomly (i.e. $p_i = 1/r$ for each i) then $W(n) = W_0 = 1/r\, \Sigma_i d_i$. If an automaton is said to learn, it must perform better than this, in which case it is called *expedient*.

DEFINITION 4.1

A learning automaton is said to be *Expedient* if

$$\lim_{n\to\infty} E[W(n)] > W_0. \tag{6}$$

The best behaviour we can expect from the automaton is defined as follows.

DEFINITION 4.2

A learning automaton is said to be *Optimal* if

$$\lim_{n\to\infty} E[W(n)] = d_m, \tag{7}$$

where

$$d_m = \max_i \{d_i\}.$$

An equivalent definition is,

$$\lim_{n\to\infty} p_m(n) = 1 \text{ (with probability 1) (w.p.1).} \tag{8}$$

DEFINITION 4.3

A learning automaton is said to be *ε-optimal* if

$$\lim_{n\to\infty} E[W(n)] > d_m - \varepsilon \tag{9}$$

can be obtained for any arbitrary $\varepsilon > 0$ by a proper choice of the parameters of the automaton.

Another definition closely related to ε-optimality is the following.

DEFINITION 4.4

A learning automaton is said to be *Absolutely Expedient* if

$$E[W(n+1)|p(n)] > W(n) \tag{10}$$

for all n, all $p_i(n) \in (0, 1)$ and for all possible sets $\{d_i\}$ $(i = 1, \ldots, r)$ except those in which all reward probabilities are equal.

It can be seen that absolute expediency implies that $W(n)$ is a submartingale and consequently $E[W(n)]$ monotonically increases in arbitrary environments. Furthermore with some additional conditions it ensures ε-optimality.

5. Learning algorithms

The basic operation in a learning automaton is the updating of action probabilities. The algorithm used for this operation is known as the learning algorithm or the reinforcement scheme (Lakshmivarahan 1981). In general terms, the learning algorithm can be represented by,

$$p(n+1) = T[p(n), \alpha(n), \beta(n)], \tag{11}$$

where T is an operator. The algorithm generates a Markov process $\{p(n)\}$. If $p(n+1)$ is a linear function of $p(n)$, the reinforcement scheme is said to be linear. Sometimes the scheme may be characterized by the asymptotic behaviour of the learning automaton using it i.e. expedient, optimal etc.

Let S_r be the unit simplex defined by

$$S_r = \{p | p = [p_1, p_2, \ldots, p_r]^T, \quad 0 \leqslant p_i \leqslant 1, \sum_{i=1}^{r} p_i = 1\}. \tag{12}$$

Then S_r is the state space of the Markov process $\{p(n)\}$. The interior of S_r is represented as S_r^0. Let $e_i = [0, 0, \ldots, 1, 0, 0]^T$ where the ith element is unity, be an r-vector. Then the set of all $e_i (i = 1, \ldots, r)$ is the set of vertices V_r of the simplex S_r.

If the learning algorithm is chosen so that $\{p(n)\}$ has absorbing states, then it is called an absorbing algorithm. Otherwise it is a nonabsorbing algorithm. The two types of algorithms have different types of behaviour.

We shall first consider a few linear algorithms which are simple to implement and analyse.

5.1 *Linear reward–penalty (L_{R-P}) scheme*

This scheme has been extensively treated in the psychology literature (Bush & Mosteller 1958). It can be described as follows.

If $\alpha(n) = \alpha_i$,

$$\begin{aligned}
p_i(n+1) &= p_i(n) + a[1 - p_i(n)], \quad \text{if } \beta(n) = 1, \\
p_j(n+1) &= (1 - a)\, p_j(n), \\
p_i(n+1) &= (1 - b)\, p_i(n), \quad \text{if } \beta(n) = 0, \\
p_j(n+1) &= b/(r-1) + (1 - b)\, p_j(n),
\end{aligned} \tag{13}$$

where $0 < a < 1$ and $b = a$. Computing the conditional expectation $E[p_i(n+1)|p(n)]$ and rearranging,

$$E[p(n+1) = A^T E[p(n)], \tag{14}$$

where A is an $(r \times r)$ stochastic matrix with elements,

$$a_{ii} = (1 - ac_i) \text{ and } a_{ij} = ac_i/(r-1).$$

A has one eigenvalue at unity and the rest in the interior of the unit circle. The asymptotic solution of (14) can be computed as the eigenvector of A corresponding to unity eigenvalue and is given by

$$\lim_{n\to\infty} E[p_i(n)] = (1/c_i) \Big/ \left[\sum_{j=1}^{r} (1/c_j)\right], \quad (i = 1, \ldots, r).$$

It follows that

$$\lim_{n\to\infty} E[W(n)] = 1 - \left\{ r \Big/ \left[\sum_{j=1}^{r} (1/c_j)\right]\right\} > \left\{\left(\sum_{j=1}^{r} d_j\right) \Big/ r\right\} = W_0,$$

and hence L_{R-P} is expedient in all stationary random environments.

5.2 *The linear reward–*inaction *(L_{R-I}) scheme*

This algorithm (Shapiro & Nagendra 1969) can be obtained by setting $b = 0$ in (13). It is sometimes called a 'benevolent' scheme as there is no updating under a penalty input.

Let

$$\Delta p_i(n) = E[p_i(n+1) - p_i(n)|p(n)]$$

$$= a\, p_i(n) \sum_j p_j(n)(d_i - d_j). \tag{15}$$

Hence,

$$\Delta p_m(n) = a\, p_m(n) \sum_j p_j(n)(d_m - d_j) > 0 \tag{16}$$

for all $p \in S_r^0$, since $(d_m - d_j) > 0$ for all $j \neq m$.

It follows that $E[p_m(n)]$ is monotonically increasing in any stationary random environment and this is certainly a desirable feature. It can also be checked that each vertex e_i of S_r is an absorbing state of the Markov process $\{p(n)\}$. Furthermore one can show that $\{p(n)\}$ converges to an element of the set of vertices V_r w.p.1 and also that $P\{\text{Lim}_{n\to\infty} p_m(n) = 1\}$ can be made arbitrarily close to unity by selecting sufficiently small 'a'. This implies that the L_{R-I} scheme is ε-optimal in all stationary random environments.

5.3 *The linear reward – ε-penalty ($L_{R-\varepsilon P}$) scheme*

The L_{R-I} scheme has the drawback of being associated with several absorbing states. If the process $\{p(n)\}$ starts at an absorbing state it continues in that state and this may not be desirable. However the scheme is ε-optimal. One can continue to enjoy

ε-optimality without the disadvantage of absorbing states by setting the parameter b in (13) to a small positive number ($b \ll a < 1$). This choice gives the $L_{R-\varepsilon P}$ scheme (Lakshmivarahan 1981).

5.4 *Absolutely expedient schemes*

Early studies of reinforcement schemes were made in a heuristic fashion. A synthesis approach led to the concept of absolutely expedient schemes (Lakshmivarahan & Thathachar 1973). The concept arose as a result of the following question: What are the conditions on the functions appearing in the reinforcement scheme that ensure desired behaviour?

The importance of absolutely expedient schemes arises partly from the fact that they represent the only class of schemes for which necessary and sufficient conditions of design are available. They can be considered as a generalization of the L_{R-I} scheme. They are also ε-optimal in all stationary random environments.

Consider a general reinforcement scheme of the following form.

If $\alpha(n) = \alpha_i$,

$$p_j(n+1) = p_j(n) - g_j(p(n)), \quad \text{when } \beta(n) = 1,$$

$$p_i(n+1) = p_i(n) + \sum_{j \neq i} g_j(p(n)), \quad (j \neq i),$$

$$p_j(n+1) = p_j(n) + h_j(p(n)), \quad \text{when } \beta(n) = 0,$$

$$p_i(n+1) = p_i(n) - \sum_{j \neq i} h_j(p(n)). \tag{17}$$

In the above g_j, h_j are continuous nonnegative functions mapping $S_r \to [0,1]$ further satisfying (for all $p \in S_r^0$)

$$0 < g_j(p) < p_j,$$

$$0 < \sum_{j \neq i} [p_j + h_j(p)] < 1, \tag{18}$$

so that $p(n+1) \in S_r^0$ whenever $p(n) \in S_r^0$.

The following theorem gives conditions on g_j, h_j for absolute expediency.

Theorem 5.1 *A learning automaton using the general algorithm* (17) *is absolutely expedient if and only if*

$$g_1(p)/p_1 = g_2(p)/p_2 = \cdots = g_r(p)/p_r$$

and

$$h_1(p)/p_1 = h_2(p)/p_2 = \cdots = h_r(p)/p_r, \tag{19}$$

for all $p \in S_r^0$.

Comments: (1) The theorem says that an absolutely expedient scheme is determined by 2 functions only. These could be designated as $\lambda(p) = g_i(p)/p_i$ and $\mu(p) = h_i(p)/p_i$, $(i = 1, \ldots, r)$. Reward--inaction schemes can be obtained by setting $\mu(p) \equiv 0$. The L_{R-I} scheme results when $\lambda(p) \equiv a, \mu(p) \equiv 0$.

(2) When d_m is unique, it can be shown that absolute expediency is equivalent to the

condition $\Delta p_m(n) > 0$ for all n, all $p \in S_r^0$ and in all stationary random environments*.
(3) Absolute expediency implies that $E[p_m(n)]$ is monotonically increasing in all stationary random environments*.
(4) Other classes of absolutely expedient schemes can be obtained by choosing different forms of reinforcement schemes. The most general scheme at present is due to Aso and Kimura (Narendra & Thathachar 1973).

5.5 *Estimator algorithms*

Estimator algorithms (Thathachar & Sastry 1985) arose from the idea that as the automaton is operating in the random environment, it is gaining information about the environment. This information can profitably be used in the updating of $p(n)$; for instance, to speed up the learning process. One such algorithm called the *pursuit algorithm* is given below.

Let

$$\hat{d}_i(n) = R_i(n)/Z_i(n), \tag{20}$$

where $R_i(n)$ = number of times reward input was obtained during the instants at which action α_i was selected up to the instant n.

$Z_i(n)$ = number of times action α_i was selected up to the instant n.
$\hat{d}_i(n)$ = estimate of d_i at n.

Let

$$\hat{d}_H(n) = \operatorname*{Max}_i [\hat{d}_i(n)].$$

Then,

$$p(n+1) = p(n) + a[e_{H(n)} - p(n)], \tag{21}$$

where $0 < a < 1$ and $e_{H(n)}$ is the unit vector with unity in position $H(n)$ and zeros in the rest.

Comments: (1) The pursuit algorithm computes the 'optimal' vector $e_{H(n)}$ at each n on the basis of the measurements up to n and moves $p(n)$ towards it by a small distance determined by the parameter a.
(2) It can be shown that the pursuit algorithm is ε-optimal in all stationary random environments. Furthermore it is an order of magnitude faster than L_{R-I} and other nonestimator algorithms.

6. Convergence and ε-optimality

The basic theoretical question in the operation of a learning automaton is the asymptotic behaviour of $\{p(n)\}$ with respect to n. It refers to the convergence of a sequence of dependent random variables.

There are two approaches to the analysis of the problem. One is based on stochastic contraction mapping principles leading to distance-diminishing operators. The other approach is through the martingale convergence theorem.

---

* Omit environments with all d_i equal.

In the study of learning algorithms two distinct types of convergence can be identified. In the first type, information on the initial state $p(0)$ is eventually lost as $p(n)$ evolves in time. An algorithm with such behaviour is said to be ergodic. Here $p(n)$ converges in distribution to a random variable (r.v.) p whose distribution is independent of $p(0)$. This is characteristic of algorithms such as L_{R-I} and $L_{R-\varepsilon P}$.

In the second type of convergence, the process $\{p(n)\}$ has a finite number of absorbing states. It can be shown that $p(n)$ converges w.p.1 to one of the absorbing states. This type of convergence is associated with L_{R-I} and other absolutely expedient schemes which are also called absorbing algorithms.

In order to show ε-optimality, one has to consider the effect of small values of the learning parameter. In ergodic algorithms, the problem can be reduced to the study of an associated ordinary differential equation which is shown to have a stable equilibrium point near the optimum value of $p(n)$. In absorbing algorithms, bounds on the probability of convergence to the optimum value of $p(n)$ are derived and it is shown that these bounds converge to 1 as the learning parameter goes to zero.

7. *Q* and *S* models

The development so far has been concerned with P models where the environment has only a binary response. Most of these results can be extended to Q and S model environments. Actually it is enough to consider S models as Q models could be regarded as particular cases of S models.

In S models the output of the environment for each action α_i is a random variable with a distribution F_i over the interval $[0, 1]$. The mean value of this distribution s_i plays the same role as the reward probability d_i in the P model,

Let

$$s_i = E[\beta(n)|\alpha(n) = \alpha_i] \tag{22}$$

This s_i is usually called the reward strength.

Each reinforcement scheme in the P model has its counterpart in the S model. For instance, the L_{R-I} scheme can be extended as follows.

The SL_{R-I} scheme

$$\begin{aligned} p_i(n+1) &= p_i(n) - a\beta(n)p_i(n), \quad \text{if } \alpha(n) \neq \alpha_i, \\ p_i(n+1) &= p_i(n) + a\beta(n)(1 - p_i(n)), \quad \text{if } \alpha(n) = \alpha_i. \end{aligned} \tag{23}$$

The scheme reduces to the L_{R-I} scheme when $\beta(n) = 0$ or 1.

Comment: When the output of the environment is a continuous variable such as the performance index of a controlled process, an S model has to be used. If $Y(n)$ is the output whose upper and lower bounds are A and B, it can be transformed to lie in the interval $[0, 1]$ by defining

$$\beta(n) = [Y(n) - B]/[A - B].$$

However, A and B are not always known in practice and may have to be estimated. The S model version of the pursuit algorithm avoids this problem as it needs only the average output due to each action up to the present instant n.

8. Hierarchical systems

When the number of actions is large, a single automaton becomes ineffective. It becomes slow as a large number of action probabilities are to be updated. One way of overcoming this complexity is to arrange a number of automata in a hierarchical system (Ramakrishnan 1982).

Figure 2 shows a number of automata arranged in 2 levels of a tree hierarchy. The first level automaton has r actions and chooses an action (say α_i) based on its action probability distribution. This action triggers the automaton A_i of the second level which in turn chooses action α_{ij} based on its own distribution. The action α_{ij} interacts with the environment and elicits a response $\beta(n)$. The reward probability is d_{ij}. The action probabilities of both A and A_i are now updated and the cycle is repeated.

The basic problem here is to find reinforcement schemes for the automata at different levels which ensure the convergence of the hierarchy to the optimal action. We shall outline an approach which results in absolute expediency.

Let the following notation be used.

$\beta(n)$ = response of the environment at n.
$p_i(n)$ = ith action probability of automaton A at n.
$p_{ij}(n)$ = jth action probability of automaton A_i at n.
d_{ij} = reward probability associated with action α_{ij} at the second level and α_i at the first level
$= P[\beta(n) = 1 | \alpha_i, \alpha_{ij}]$
r_i = number of actions of A_i.

Let the first level automaton A use a reward–inaction absolutely expedient scheme as follows.

α_i = action selected at n.

$$\left.\begin{aligned} p_i(n+1) &= p_i(n) + \lambda(p(n))[1 - p_i(n)] \\ p_j(n+1) &= p_j(n)[1 - \lambda(p(n))] \end{aligned}\right\} \text{if } \beta(n) = 1$$

$$p_k(n+1) = p_k(n), \quad (i = 1, \ldots, r), \quad \text{if } \beta(n) = 0. \tag{24}$$

Similarly let the second level updating be as follows.

α_{ij} = action selected at n.

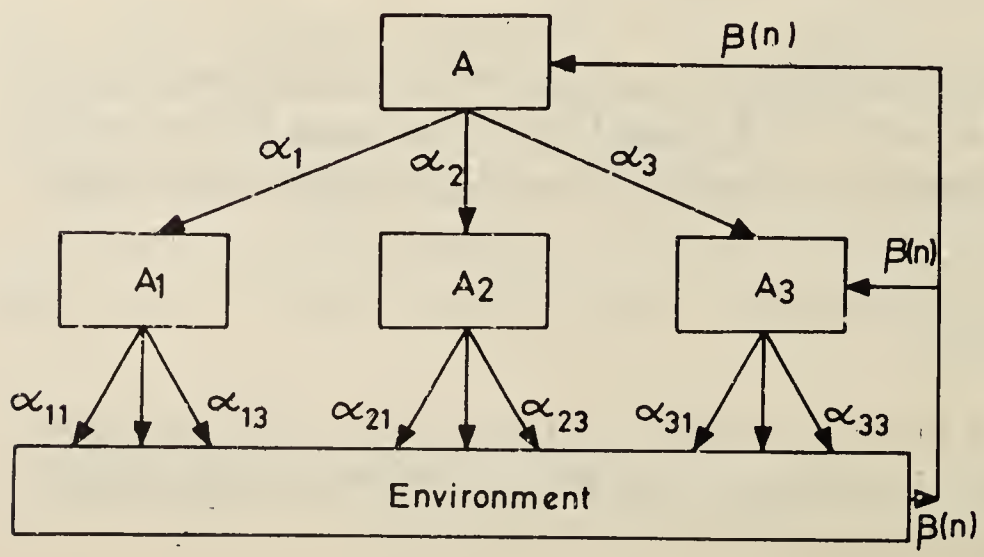


Figure 2. A 2-level hierarchical system.

Automaton A_i

$$p_{ij}(n+1) = p_{ij}(n) + \lambda_i(p(n))(1 - p_{ij}(n)), \quad \text{if } \beta(n) = 1,$$
$$p_{ik}(n+1) = p_{ik}(n)[1 - \lambda_i(p(n))], \quad (k \neq j). \tag{25}$$

Automaton $A_k (k \neq i)$

No updating

In terms of the above quantities there is a simple relationship which ensures absolute expediency of the hierarchy, i.e. the hierarchy is equivalent to a single automaton which is absolutely expedient.

Theorem 8.1. *The hierarchical system described by* (24), (25) *is absolutely expedient, if and only if*

$$\lambda_i(p(n)) = [\lambda(p(n))]/[p_i(n+1)] \tag{26}$$

for all n, *all* $p(n) \in S_r^0$ *and all* $i = 1, \ldots, r$.

Comments: (1) The results stated in theorem 8.1 can be extended to any number of levels. Basically one has to divide the λ of each automaton by the connecting action probability of the previous level at $(n+1)$.
(2) The division operation need not cause concern as it will not lead to division by zero w.p.1.
(3) Number of updatings is reduced from r^2 to $2r$ in a 2-level hierarchy and r^N to Nr in an N-level hierarchy.

9. Team of learning automata

An alternative manner in which a complex learning problem can be handled is through a number of learning automata arranged to form a team. Each member of the team has the same goal. An example of such a team is in a game with identical payoffs. Here each automaton of the team $A_1, A_2, \ldots, A_N$ has a finite number of actions to choose from. At an instant n, each automaton chooses one action following which the environment gives out the same payoff to each member of the team. The automata update their action probabilities and choose new actions again. The process repeats. The objective of all the automata is to maximize the common payoff. The set up is shown in figure 3 (Ramakrishnan 1982).

The automata can operate in total ignorance of other automata, in which case it is a decentralized game. Alternatively some kind of information transfer can take place among the automata so as to improve the speed of operation or other characteristics of the game.

Let the action set of automaton A_k be α^k and the action probability vector at instant n be $p(k, n)$ $(k = 1, 2, \ldots, N)$. Also let each automaton operate according to a generalized nonlinear reward–inaction scheme described below.

Let the action chosen by A_k at n be $\alpha_{i_k}^k$. Then for each $k = 1, 2, \ldots, N$,

$$p(k, n+1) = p(k, n) + (G^k)^T e_{i_k}(k), \quad \text{for } \beta(n) = 1,$$
$$p(k, n+1) = p(k, n). \tag{27}$$

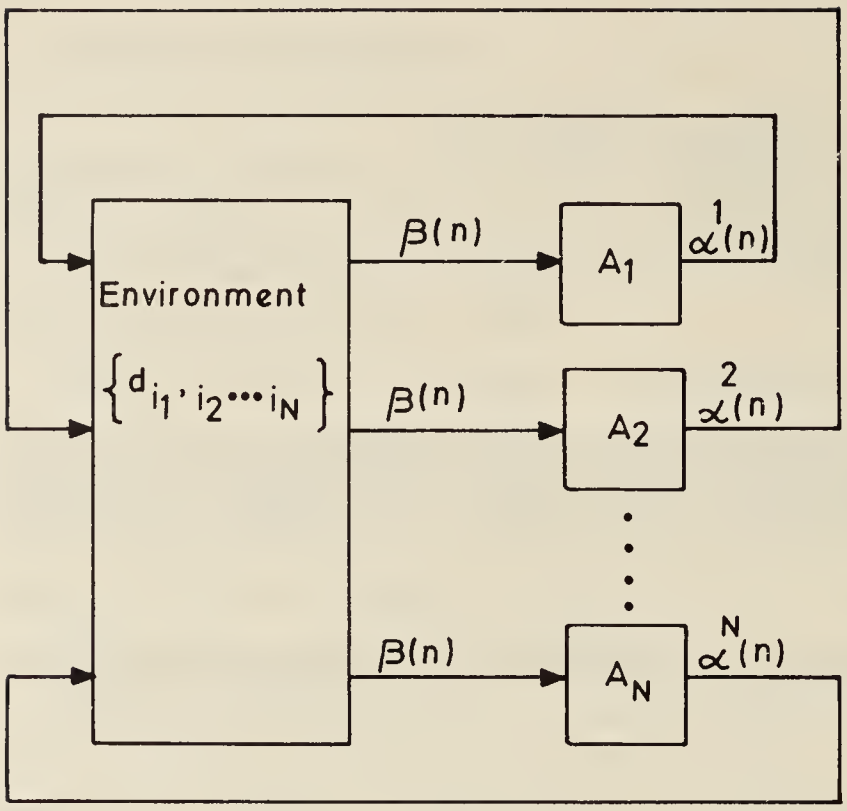


Figure 3. Automata game with common pay-off.

where

(i) $e_{ik}(k)$ is the $(r_k \times 1)$ unit vector having unity for the i_kth element and the rest of the elements are zero.

(ii) G^k stands for G^k $(p(k,n))$ and $G^k(p)$ is an $r_k \times r_k$ matrix with its (i,j)th element $g^k_{ij}(p)$ $(j \neq i)$ and (i,i)th element $g^k_{ii}(p) = -\Sigma_{j \neq i} g^k_{ij}(p)$.

The environment is described by a hypermatrix of reward probabilities defined by

$$d_{i,i_2,\ldots,i_N} = P[\beta(n) = 1 | \alpha^k_{i_k} \quad \text{chosen by } A_k \text{ at } n,\ k = 1,\ldots,N]. \tag{28}$$

For optimality, each automaton A_k should converge to the action $\alpha^k_{m_k}$ w.p.1, where

$$d_{m1m2},\ldots,m_N = \max_{i,\ldots,i_N} [d_{i,i_2,\ldots,i_N}]. \tag{29}$$

A notion related to absolute expediency can be defined as follows.

DEFINITION 9.1

A learning algorithm for the game of N automata with identical payoffs is said to be absolutely monotonic if

$$W(n) = E[W(n+1) - W(n) | p(k,n), k = 1,\ldots,N] > 0 \tag{30}$$

for all n, all $p(k,n) \in S^0_{r_k}$ and all possible game environments.

Comment: Absolute monotonicity ensures monotonic increase of $E[W(n)]$ in arbitrary game environments.

The class of absolutely monotonic learning algorithms can be characterized as follows.

Theorem 9.1. *Necessary and sufficient conditions for the learning algorithm* (27) *to be absolutely monotonic are given by*

$$p^T(k,n) G^k(p(k,n)) = 0 \tag{31}$$

for all $k = 1,\ldots,N$, *all* n *and all* $p(k,n) \in S^0_{r_k}$.

Comment: It can be checked that absolutely expedient algorithms of the reward–inaction type are also absolutely monotonic. Thus the L_{R-I} scheme is absolutely monotonic. It appears difficult to include general penalty terms.

A team using an absolutely monotonic learning algorithm may not be ε-optimal. To appreciate the difficulty consider a 2-player game where each automaton player has 2-actions. The game environment can be represented by the matrix of reward probabilities D.

$$D = [d_{ij}] = \begin{bmatrix} 0{\cdot}8 & 0{\cdot}1 \\ 0{\cdot}3 & 0{\cdot}6 \end{bmatrix}.$$

Here the rows correspond to actions of automaton A_1 and columns to those of A_2. If both the automata choose the first action, the probability of getting a reward is 0·8.

Looking at the matrix D, one can identify 0·8 and 0·6 as local maxima as they are the maximum elements of their row and column. However, 0·8 is the global maximum and we would like the automata A_1 and A_2 to converge to their first actions as this leads to the highest expected payoff.

The problem with absolutely monotonic schemes is that they could converge to any of the local maxima. Only in the case of a single local maximum do they lead to ε-optimality.

In spite of the above limitation, the above result is of fundamental importance in the decentralized control of large systems. It indicates that simple policies used independently by an individual decision maker could lead to desirable group behaviour.

Estimator algorithms could be used to overcome the difficulty associated with a team converging to a local maximum. It can be shown that these algorithms converge to the global maximum in the sense of ε-optimality. The pursuit algorithm applied to the game with common payoff can be stated as follows (Thathachar & Sastry 1985; Mukhopadhyay & Thathachar 1989).

1. $A_1, A_2, \ldots, A_k, \ldots, A_N$ are the N automata players;
2. the kth player A_k has r_k strategies (actions), $k = 1, 2, \ldots, N$;
3. $\{\alpha_1^k, \alpha_2^k, \ldots, \alpha_{r_k}^k\}$ is the set of actions of A_k;
4. $p(k,n) = [p_1(k,n), p_2(k,n), \ldots, p_{r^k}(k,n)]^T$ is the action probability vector of A_k at instant n;
5. $\alpha^k(n) =$ action selected by A_k at n.
6. $\beta(n) =$ payoff at n(0 or 1) common to all players;
7. D is the N-dimensional hypermatrix with elements

$$d_{i_1, i_2, \ldots, i_N} = P[\beta(n) = 1 | \alpha^k(n) = \alpha_{i_k}^k,\ k = 1, \ldots, N];$$

8. $d_{m_1, m_2, \ldots, m_N} = \max_{i_1, i_2, \ldots, i_N}(d_{i_1, i_2, \ldots, i_N})$;
9. each A_k maintains an estimate of D in $\hat{D}(n)$ as indicated in the next algorithm;
10. $E^k = [E_1^k, E_2^k, \ldots, E_{r_k}^k]^T, \quad k = 1, \ldots, N,$

and

$$E_j^k = \max_{\substack{i_s,\ 1 \leq s \leq N \\ s \neq k}} \{d_{i_1, i_2, \ldots, i_{k-1} j i_{k+1}, \ldots, i_N}\}$$

$$\hat{E}_j^k(n) = \max_{\substack{i_s, i \leq s \leq N \\ s \neq k}} \hat{d}(n)_{i_1, i_2, \ldots, i_{k-1} j i_{k+1}, \ldots, i_N}\}$$

is an estimate of E_j^k at n;

11. H_k is a random index such that

$$\hat{E}^k_{H_k}(n) = \max_i \{\hat{E}^k_i(n)\}.$$

9.1 *The pursuit algorithm*

Let $\alpha^k(n) = \alpha^k_i$ $(k = 1, \ldots, N)$. The updating of $p(k, n)$ is as follows,

$$p(k, n+1) = p(k, n) + a[e_{H_k} - p(k, n)]. \tag{32}$$

In the above, e_{H_k} is the unit vector with unity in the H_kth position and zero in the rest. The parameter a satisfies $0 < a < 1$. The reward probability estimates are updated as follows.

$$\begin{aligned}
R_{i_1 i_2 \cdots i_N}(n+1) &= R_{i_1 i_2 \cdots i_N}(n) + \beta(n), \\
R_{j_1 j_2 \cdots j_N}(n+1) &= R_{j_1 j_2 \cdots j_N}(n), \quad (j_k \neq i_k), \\
Z_{i_1 i_2 \cdots i_N}(n+1) &= Z_{i_1 i_2 \cdots i_N}(n), \quad +1 \\
Z_{j_1 j_2 \cdots j_N}(n+1) &= Z_{j_1 j_2 \cdots j_N}(n), \quad (j_k \neq i_k).
\end{aligned}$$

In the above $Z_{i_1 i_2 \cdots i_N}(n)$ represents the count of the number of times the action set $\{\alpha^1_{i_1}, \alpha^2_{i_2}, \ldots, \alpha^N_{i_N}\}$ has been selected up to the instant n. $R_{i_1 i_2 \cdots i_N}(n)$ is the number of times the reward was obtained at the instants the same action set was selected up to instant n. Thus the estimates of the reward probabilities can be computed as

$$\hat{d}_{j_1 j_2 \cdots j_N}(n+1) = [R_{j_1 j_2 \cdots j_N}(n)]/[Z_{j_1 j_2 \cdots j_N}(n)]. \tag{33}$$

The convergence result for the team using this algorithm can be stated as follows.

Theorem 9.2 *In every stationary randon game environment, a team of learning automata playing a game with identical payoffs using the pursuit algorithm is ε-optimal. That is, given any $\varepsilon > 0$, $\delta > 0$, there exist $a^* > 0$, $n_0 < \infty$ such that*

$$P[|p_{m_k}(k, n) - 1| < \varepsilon] > 1 - \delta \tag{34}$$

for all $n > n_0, 0 < a < a^*$ and $k = 1, 2, \ldots, N$.

Outline of Proof. The proof of the theorem depends on 3 main ideas.

(1) If the learning parameter a is chosen to be sufficiently small, all action N tuples are selected any specified number of times with probability close to unity.
(2) Under the above conditions, $\hat{D}(n) \to D$ and $\hat{E}(n) \to E$.
(3) For each of the automata A_k the game playing algorithm is equivalent to a single pursuit algorithm with $\hat{E}^k_j$ taking the role of the estimate of the reward probability. Hence each automaton converges to the optimal action α^k_{mk} in the sense indicated.

Comment: In a practical learning problem the parameter a has to be chosen carefully. Too small a value will make learning too slow to be useful. Too large a value may speed up convergence but may also lead to convergence to the wrong actions.

10. Application to pattern recognition

Let patterns represented by m-dimensional vectors from a feature space X belong to one of two classes w_1 and w_2. The class conditional densities $f(x/w_i)$ and the prior probabilities $P(w_i)$ $(i = 1, 2)$ are assumed to be unknown. Only a set of sample patterns with known classification is available for training the pattern recognizer (Thathachar & Sastry 1987).

To classify new patterns, a discriminant function $g(x): R^m \to R$ is defined such that the following rule can be used.

$$\begin{aligned} &\text{If } g(x) \geqslant 0, \quad \text{decide } x \in w_1, \\ &\text{if } g(x) < 0, \quad \text{decide } x \in w_2. \end{aligned} \tag{35}$$

The task is to determine $g(x)$ which minimizes the probability of misclassification.

It is known that the Bayes' decision rule,

$$x \in w_1, \text{if } P(w_1/x) \geqslant P(w_2/x), \tag{36}$$

minimizes the probability of misclassification. Thus the optimum discriminant function is given by

$$g_{\text{opt}}(x) = P(w_1/x) - P(w_2/x), \tag{37}$$

and must be determined using the training samples.

Let a known form of discriminant function with N parameters $\theta_1, \theta_2, \ldots, \theta_N$ be assumed.

$$g(x) = h(\theta_1, \theta_2, \ldots, \theta_N, x). \tag{38}$$

Then with the decision rule (35), the probability of correct classification for a given $\theta = (\theta_1, \ldots, \theta_N)$ is given by

$$J(\theta) = P(w_1)P(g(x) \geqslant 0 | x \in w_1) + P(w_2)P(g(x) < 0 | x \in w_2). \tag{39}$$

For a sample pattern x, let

$$\begin{aligned} L(x) &= 1, \quad \text{if } x \text{ is properly classified,} \\ &= 0, \quad \text{otherwise.} \end{aligned} \tag{40}$$

Then,

$$J(\theta) = E[L(x)], \tag{41}$$

and hence maximizing $E[L(x)]$ maximizes the probability of correct classification.

The pattern classification problem can now be posed as a game of N automata $A_1, A_2, \ldots, A_N$ each of which chooses a parameter θ_i $(i = 1, \ldots, N)$ to maximize the identical payoff $J(\theta)$. Hence the results of §9 can be applied provided each parameter θ_i is discretized and consequently belongs to the finite action set of A_i. Further, since complete communication between the automata can be assumed, estimator algorithms can be used to speed up convergence rates.

The training of the classifier proceeds as follows. Let $\theta_i \in \alpha^i$ where α^i is the finite action set of A_i defined by

$$\alpha^i = \{\alpha_1^i, \alpha_2^i, \ldots, \alpha_r^i\}.$$

In the game, each automaton A_i chooses a particular action (and hence a value of θ_i), and this results in a classifier with a discriminant function $g(x)$. Any sample pattern

is classified according to rule (35) and $L(x)$ is determined according to (36). This $L(x)$ forms the payoff for all the automata that update their action probabilities following an estimator algorithm. If the set of actions selected at any stage is $\{\alpha_{i_1}^1, \alpha_{i_2}^2, \ldots, \alpha_{i_N}^N\}$ the corresponding reward probability is $d_{i_1,i_2,\ldots,i_N}$. For using estimator algorithms it is necessary to estimate $d_{i_1,i_2,\ldots,i_N}$ and update the estimate with each pattern. As estimator algorithms are ε-optimal, the parameter converges to its optimal value with arbitrary accuracy when a proper selection of the learning parameter is made in the algorithm.

Although L_{R-I} and other absolutely monotonic algorithms can also be used for updating action probabilities, estimator algorithms are preferred for two reasons. The estimator algorithms invariably converge to the global optimum whereas the former converge only to local maxima. Furthermore the nonestimator algorithms are very slow and need substantially larger training sets.

In conclusion, the automata team learns the optimal classifier under the following assumptions:

(1) The form of the optimal discriminant function is contained in the functional form of $g(x)$ chosen.
(2) The optimal values of the parameters are elements of the sets $\alpha^i (i = 1, 2, \ldots, N)$.

Even when these assumptions are not satisfied, the team of automata would learn the best classifier among the set of classifiers being considered. By choosing finer parameter sets, successively better approximations to the optimal classifier can be obtained.

Example: Let $X = [x_1, x_2]^T$ be the feature vector with independent features. The class conditional densities

$$f(x_1|w_1) \text{ and } f(x_2|w_1) \text{ are uniform over } [1, 3],$$

$$f(x_1|w_2) \text{ and } f(x_2|w_2) \text{ are uniform over } [2, 4].$$

A discriminant function of the form

$$g(X) = 1 - (x_1/\theta_1) - (x_2/\theta_2),$$

is assumed with a range of parameters [0, 10]. This interval is discretized into 5 values.

The discriminant function learned by a 2-automaton team is

$$g(X) = 1 - (x_1/5) - (x_2/5),$$

which is optimal.

The average number of iterations for the optimum action probabilities to converge to 0·95 is 930. The team converged to the optimal discriminant function in 8 out of 10 runs. In the other two runs it converges to a line on the x_1–x_2 plane close to the optimal.

11. Other applications

There are several areas of computer and communication engineering where learning automata have been found to be significantly useful. A few of these are outlined below.

There has been a great deal of interest in applying adaptive routing to communication

networks. Since these networks generally involve large investments, even a small improvement in their working efficiency results in appreciable savings in operational expenses. Inherently, communication networks are such that the volume as well as the pattern of traffic vary over wide ranges. Unusual conditions arise because of component failures and natural disasters. These considerations lead to the necessity of using learning control algorithms to achieve improved performance under uncertain conditions.

In circuit switched networks (such as telephone networks), a learning automaton is used at a node to route the incoming calls to other nodes (Narendra & Mars 1983). Typically if a call from a source node i, destined for node j is at node k, an automaton A_{ij}^k is used to route the calls at node k. The actions of the automaton are either the links connected to node k or the sequence in which the connecting links are to be tried. The environmental response is the information whether the call reached the destination or not. Typically $L_{R-\varepsilon P}$ and L_{R-P} schemes have been used in adaptive routing. It has been observed that when $L_{R-\varepsilon P}$ schemes are used, the blocking probabilities along the route at each node are equalized. Similarly blocking rates are equalized when L_{R-P} schemes are used for routing.

It has been generally concluded that automata at the various nodes act in such a manner as to equalize the quality of service at the different nodes as measured by blocking probabilities corresponding to different loads. In the presence of abnormal operating conditions such as link failure, the automata algorithms result in significantly reduced blocking probabilities and node congestion provided additional capacity is available in the network.

Similar remarks apply to the use of automata for routing messages in packet-switched networks also (Mason & Gu 1986, pp. 213–228).

There are also some queueing problems such as task scheduling in computer networks where automata play a useful role. Learning automata have been shown to provide a good solution to the image data compression problem by selecting a proper code to transmit an image over a communication channel (Narendra & Thathachar 1989). Another prominent application is in the consistent labelling problem (Thathachar & Sastry 1986) where a team of automata provide consistent labels to objects in a scene. A consequence of this is an efficient solution of low-level vision problems such as stereocorrespondence and shape matching (Sastry *et al* 1988; Banerjee 1989).

12. Generalized learning automata

Collectives of learning automata such as hierarchies and teams have been seen to be effective in handling complex learning problems such as pattern recognition and also in a number of other applications. These are examples of certain types of interconnection of a number of automata and could be regarded as artificial neural networks where the learning automaton plays the role of an artificial neuron. The terminology seems appropriate because of the parallel and distributed nature of the interconnection and the perceptual tasks the collectives can perform. However, the nature of interconnection is somewhat different from that usually assumed in the literature, where the output of each unit forms one of the inputs to several other units. The only input to the learning automaton is the response of the environment and there is no other input from other automata. Thus a generalization of the automaton appears necessary to provide the type of interconnection normally envisaged.

The structure of a learning automaton can be generalized in two directions (Williams 1988). The first generalization is in parametrization of the state space of the automaton. Instead of the simplex S_r, one could have an arbitrary state space θ and use a mapping $\theta \rightarrow S_r$ to generate the action probabilities. For instance, in the case of a two-action automaton whose state space θ is the real line, one could use the mapping

$$\psi(\theta) = 1/(1 + e^{-\theta}) = p_1$$

to compute p_1, the probability of selection of α_1. Functions suitable for higher dimensions are harder to identify.

The second generalization one could consider is to allow the automata to have another input apart from the response of the environment. The idea is that the optimal action of the automaton may differ in differing contexts. The second input is meant for representing the context and is called the context input or context vector. When the context input is a constant, the structure of the automaton reduces to the classical one.

A parametrized-state learning automaton with context input is called a generalized learning automaton (Narendra & Thathachar 1989). It has also been called an associative stochastic learning automaton (William 1988) as it is trying to learn which actions are to be associated with which context inputs. One could think of the decision space divided into a number of regions where each region is associated with a context input and the task of the automaton is to determine the best action associated with each context input.

Generalized learning automata can be connected together to form networks which interact with the environment. Here the actions of individual automata are treated as outputs and these may in turn serve as context inputs to other automata or as outputs of the network. Some automata may receive context inputs from the environment. The environment also produces the usual response (also called reinforcement signal) which is common to all the automata and is a globally available signal. In this respect the network shares a property with the game with identical payoff. A schematic diagram of such a network is shown in figure 4.

The following operations take place in the network over a cycle.

1. The environment picks an input pattern (i.e. set of context inputs) for the network randomly. The distribution of the inputs is assumed to be independent of prior events within the network or environment.
2. As the input pattern to each automaton becomes available, it picks an action randomly according to the action probability distribution corresponding to that particular input. This 'activation' of automata passes through the network towards the 'output side'.
3. After all the automata at the output side of the network have selected their actions, the environment picks a reinforcement signal randomly according to a distribution corresponding to the particular network output pattern chosen and the particular context input to the network.
4. Each automaton changes its internal state according to some specified function of its current state, the action just chosen, its context input and the reinforcement signal.

While the above concept of a network of generalized learning automata is conceptually simple and intuitively appealing, there are very few analytical results

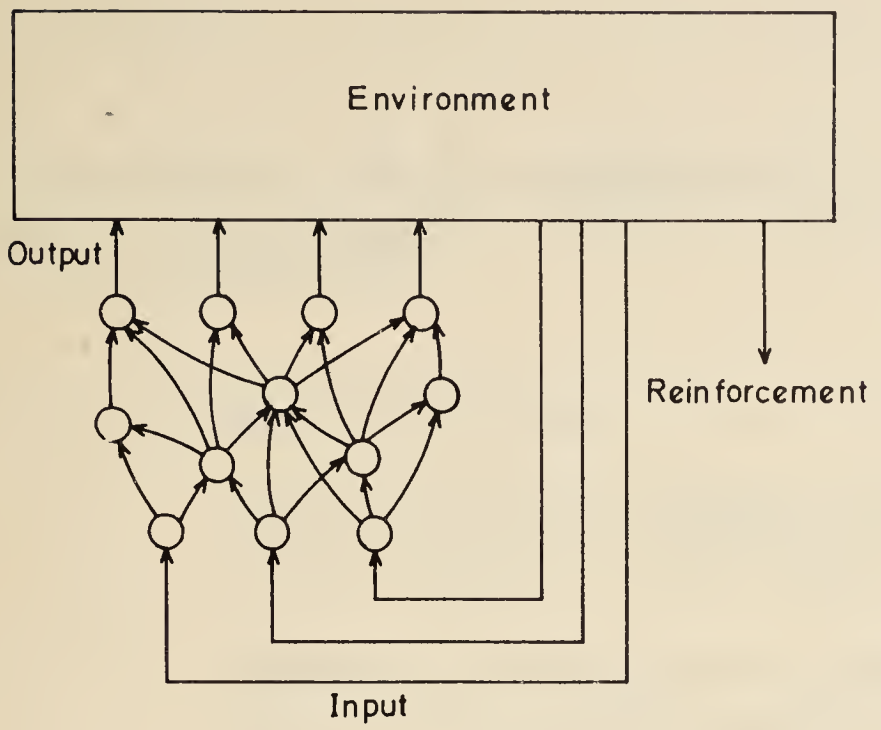


Figure 4. Network of generalized learning automata.

concerning it. Even the available results are not as complete as those of hierarchies or games of learning automata. New techniques may be needed for the analysis of such networks and for bringing out the tasks for which they are well suited.

References

Banerjee S 1989 *On stochastic relaxation paradigms for computational vision*, PhD thesis, Electrical Engineering Department, Indian Institute of Science, Bangalore

Bush R R, Mosteller F 1958 *Stochastic models for learning* (New York: John Wiley and Sons)

Lakshmivarahan S 1981 *Learning algorithms – theory and applications* (New York: Springer Verlag)

Lakshmivarahan S, Thathachar M A L 1973 Absolutely expedient learning algorithms for stochastic automata. *IEEE Trans. Syst., Man Cybern.* SMC-3: 281–283

Mason L G, Gu X D 1986 Learning automata models for adaptive flow control in packet-switching networks. In *Adaptive and learning systems* (ed.) K S Narendra (New York: Plenum)

Mukhopadhyay S, Thathachar M A L 1989 Associative learning of Boolean functions. *IEEE Trans. Syst. Man Cybern.* SMC-19: 1008–1015

Narendra K S, Mars P 1983 The use of learning algorithms in telephone traffic routing – A methodology, *Automatica* 19: 495–502

Narendra K S, Thathachar M A L 1989 *Learning automata – an introduction* (Englewood Cliffs, NJ: Prentice Hall)

Ramakrishnan K R 1982 *Hierarchical systems and cooperative games of learning automata*, PhD thesis, Indian Inst. Sci. Bangalore

Sastry P S, Banerjee S, Ramakrishnan K R 1988 A local cooperative processing model for low level vision. *Indo-US Workshop on Systems and Signal Processing* (ed.) N Viswanadham (New Delhi: Oxford & IBH)

Shapiro I J, Narendra K S 1969 Use of stochastic automata for parameter self-optimization with multimodal performance criteria. *IEEE Trans. Syst. Sci. Cybern.* SSC-5: 352–360

Thathachar M A L, Sastry P S 1985 A new approach to the design of reinforcement schemes for learning automata. *IEEE Trans. Syst., Man Cybern.* SMC-15: 168–175

Thathachar M A L, Sastry P S 1986 Relaxation labeling with learning automata. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-8: 256–268

Thathachar M A L, Sastry P S 1987 Learning optimal discriminant functions through a cooperative game of automata. *IEEE Trans. Syst., Man Cybern.* SMC-17: 73–85

Tsetlin M L 1973 *Automaton theory and modeling of biological systems* (New York: Academic Press)

Williams R J 1988 Toward a theory of reinforcement – learning connectionist systems, Technical Report, NU-CCS-88-3, College of Computer Science, Northeastern University, Boston, MA

An analysis of the equilibria of neural networks with linear interconnections

M VIDYASAGAR

Centre for Artificial Intelligence and Robotics, Raj Bhavan Circle, High Grounds, Bangalore 560 001, India

Abstract. In this paper, we analyse the equilibria of neural networks which consist of a set of sigmoid nonlinearities with linear interconnections, *without* assuming that the interconnections are symmetric or that there are no self-interactions. By eliminating these assumptions, we are able to study the effects of imperfect implementation on the behaviour of Hopfield networks. If one views the neural network as evolving on the open n-dimensional hypercube $H = (0, 1)^n$, we have the following conclusions as the neural characteristics become steeper and steeper: (i) There is at most one equilibrium in any compact subset of H, and under mild assumptions this equilibrium is unstable. In fact, the dimension of the stable manifold of this equilibrium is the same as the number of eigenvalues of the interconnection matrix with negative real parts. (ii) There might be some equilibria in the faces of H, and under mild conditions these are always unstable. Moreover, it is easy to compute the dimension of the stable manifold of each such equilibrium. (iii) A systematic procedure is given for determining which corners of the hypercube H contain equilibria, and it is shown that all equilibria in the corners of H are asymptotically stable.

Keywords. Neural networks; linear interconnections; sigmoid non-linearities.; stability of equilibria.

1. Introduction

Recently there has been a great deal of interest in artificial neural networks, especially of the so-called Hopfield type (Hopfield 1982, 1984; Hopfield & Tank 1985; Tank & Hopfield 1986). These consist of an interconnection of so-called sigmoid non-linearities which represent the neurons or the switching elements, which are then interconnected through *linear* gains. The equations representing the dynamic behaviour of such networks are of the form

$$C_i \dot{u}_i = -(1/R_i)u_i + \sum_{j=1}^{n} t_{ij} V_j + I_i, \quad i = 1, \ldots, n, \tag{1}$$

where n is the number of neurons, V_i is the neural current and u_i is the neural voltage; I_i is the external current input to the ith neuron, C_i is the membrane capacitance, R_i

is the neural resistance, and t_{ij} is the interconnection term. In Hopfield networks, two further assumptions are made, namely: (i) $t_{ii} = 0$ for all i (no self-interactions), and (ii) $t_{ij} = t_{ji}$ for all i, j (symmetric interactions). If one forms the *energy function*

$$E = -\frac{1}{2}\sum_{i=1}^{n}\left[\left(\sum_{j=1}^{n} t_{ij}V_j + 2I_i\right)V_i - (2/R_i)\int_0^{V_i} g_i^{-1}(V)\mathrm{d}V\right], \tag{2}$$

then it is straightforward to show that $\dot{E} \leqslant 0$ along the solution trajectories of the network. As a consequence (Hirsch 1987), all solution trajectories of the network approach an equilibrium. It is claimed (see e.g. Hopfield & Tank 1985 and Tank & Hopfield 1986) that neural networks can be used to solve a wide variety of problems in engineering and science by recasting them as minimization problems.

In the analyses of Hopfield (1982, 1984), Hopfield & Tank (1985), Tank & Hopfield (1986) and Hirsch (1987), the assumption that the neural interactions are symmetric is crucial, since otherwise one cannot form the energy function E of (2). The assumption that there are no self-interactions can be dispensed with in some, but not all, of the analyses. In a practical implementation of a neural network, it is often difficult to assure that the interactions are symmetric, since this often requires guaranteeing that two physical quantities (such as resistances or the gains of operational amplifiers) are *exactly* equal. Because of this, the behaviour of artificial neural networks constructed in the laboratory does not always correspond to that of ideal neural networks.

The objective of the present paper is to analyse the number, location, and stability behaviour of neural networks described by (1), *without* the assumptions of no self-interactions and symmetric interactions. On the other hand, since our interest is in studying the effects of imperfect implementations of Hopfield nets, it is assumed that the interconnection matrix is "nearly" symmetric. In other words, it is assumed that there is a *nominal* interconnection matrix T_0 which *is* symmetric, and that the actual interconnection matrix T lies in some ball of radius ε centred at T_0. Thus all the results stated here are of the form "For sufficiently small ε, something or the other is true". It shown that, even in the absence of these assumptions, it is possible to deduce a fairly complete picture of the dynamics of such networks. Specifically, it is shown that if one considers the neural network as evolving on the open hypercube $H = (0, 1)^n$ in the "V"-space, then we have the following conclusions as the neural characteristics become steeper and steeper: (i) There is at most one equilibrium in any compact subset of H, and under mild assumptions this equilibrium is unstable. In fact, the dimension of the stable manifold of this equilibrium is the same as the number of eigenvalues of the interconnection matrix with negative real parts. (ii) There might be some equilibria in the faces of H, and under mild conditions these are always unstable. Moreover, it is easy to compute the dimension of the stable manifold of each such equilibrium. (iii) A systematic procedure is given for determining which corners of H contain equilibria, and it is shown that all equilibria in the corners of H are asymptotically stable.

2. Preliminaries

In this section the various assumptions made throughout the paper are briefly summarized.

The input-output relationship of the ith neuron is given by the sigmoid function

$$V_i = g_i(\lambda u_i), \tag{3}$$

where g_i is a given sigmoid function and λ is a scaling constant. The only assumptions made on the sigmoid function are the following.

Assumptions on the sigmoid nonlinearities: $g_i(x)$ is continuously differentiable, strictly increasing, and $g_i(x) \to 1$ as $x \to \infty$, $g_i(x) \to 0$ as $x \to -\infty$. Further, $xg_i'(x) \to 0$ as $|x| \to \infty$.

The assumptions about g_i are quite standard. The assumptions about g_i' are almost a consequence of the fact that $g_i(x)$ has a definite limit as $|x| \to \infty$. Since the function $1/x$ is not integrable over any infinite interval, it follows that

$$g_i(x) \to 1 \text{ as } x \to \infty \Rightarrow \liminf xg_i'(x) = 0 \text{ as } x \to \infty, \tag{4}$$

and similarly as $x \to -\infty$. Note that commonly used sigmoid functions such as $(1 + \tanh x)/2$ and $1/(1 + e^{-x})$ satisfy these assumptions. As $\lambda \to \infty$, the sigmoid becomes steeper and steeper and eventually approaches a switching function. Note that each neuron can have a different switching function, but for simplicity it is assumed that all neurons have the same scaling constant. Now define

$$b_{ij} = t_{ij}/C_i,\ y_i = I_i/C_i,\ i = 1, \ldots, n. \tag{5}$$

As mentioned in § 1, no assumptions are made on the symmetry or otherwise of the *actual* interconnection matrix T. But there is a *nominal* interconnection matrix T_0 which is symmetric and satisfies the following conditions:

Assumptions on the nominal interconnection matrix: The matrix T_0 and all of its principal submatrices of size 2×2 or larger are hyperbolic, *i.e.* none of their eigenvalues has a zero real part. Every principal submatrix of T_0 of size 2×2 or larger has at least one eigenvalue with positive real part.

All of the matrices proposed by Tank and Hopfield satisfy these assumptions. Note that, if the nominal interconnection matrix T_0 has zero diagonal elements, then the first assumption implies the second. This is because the trace of a matrix is equal to the sum of its eigenvalues. Thus if the trace of T_0 is zero, and it has no eigenvalues on the imaginary axis, then it must have some eigenvalues with positive real part and others with negative real part. Another important point to note is that the above assumptions are *structurally stable* in the sense that if the nominal (symmetric) interconnection matrix T_0 satisfies them, so do all (nonsymmetric) matrices sufficiently close to T_0.

Define H to be the open hypercube $(0, 1)^n$, and $\bar{H}$ to be the closed hypercube $[0, 1]^n$. The symbol $\mathbf{b}$ denotes the binary set $\{0, 1\}$, and $\mathbf{b}^n$ denotes the set of n-dimensional binary vectors. Note that the set $\mathbf{b}^n$ consists precisely of the 2^n corners of the hypercube $\bar{H}$. The *faces* of the hypercube $\bar{H}$ consist precisely of those vectors $x \in \bar{H}$ with the property that $x_i \in \mathbf{b}$ for some but not all values of i. In other words, a face of $\bar{H}$ is a set of the form

$$\{x \in \bar{H}\colon x_i \in \mathbf{b}\ \forall i \in I,\ x_i \in [0, 1]\ \forall i \in J\}, \tag{6}$$

where I, J is a nontrivial partition of the set $\{1, \ldots, n\}$.

The idea is to analyse the dynamical behaviour of the neural network for different

values of the input vector y. It turns out that there are some "exceptional" values of y for which the behaviour is difficult to predict. But this exceptional set is quite small. A subset of R^n is said to be *of type* LV if it is contained in a finite union of linear varieties, each of dimension less than n. Note that a set of type LV has measure zero as a subset of H.

Among other things, we are interested in where the equilibria of (1) can lie as the sigmoid gain $\lambda \to \infty$. Three types of equilibria are identified. (1) If $V \in \bar{H}$ is an equilibrium and $V_i \in (0, 1) \forall i$, then the equilibrium is said to be in the *interior* of $\bar{H}$. (2) If all components of V approach either 0 or 1 as $\lambda \to \infty$, then the equilibrium is said to be in a *corner* of $\bar{H}$. (3) If some components of V approach 0 or 1 as $\lambda \to \infty$ while others approach some value in (0, 1), then the equilibrium is said to be in a *face* of $\bar{H}$.

3. Motivation – single-neuron case

Much of what happens in a neural network as the neuron characteristic becomes steeper and steeper can be understood by studying the behaviour of a single neuron. In this case, the neuron dynamics are described by

$$\dot{u} = -u/\alpha + bg(\lambda u) + y, \tag{7}$$

where

$$\alpha = R_1 C_1,\ b = t_{11}/C_1,\ y = I_1/C_1. \tag{8}$$

So the equilibria of this network are at the solutions of

$$(u/\alpha) = bg(\lambda u) + y. \tag{9}$$

Figure 1 shows where the solutions of this equation can lie as $\lambda \to \infty$ when $b > 0$, while figure 2 does the same when $b < 0$. These figures show that as $\lambda \to \infty$, there can be two types of equilibria. First, those where u_{eq} approaches a finite number, and V_{eq} approaches 0 if $u_{eq} < 0$ and 1 if $u_{eq} > 0$; these types of equilibria are labelled as type A in figures 1 and 2. Second, those where $u_{eq} \to 0$ but V_{eq} approaches a number strictly between 0 and 1; this type of equilibrium is labelled as type B in figure 1. In §6 we shall see that, in the case of multiple neurons, it is possible for *some components* of u_{eq} to approach a nonzero value while the remaining components approach zero; in such a case, some components of V_{eq} approach 0 or 1 while the remaining components approach a value strictly between 0 and 1.

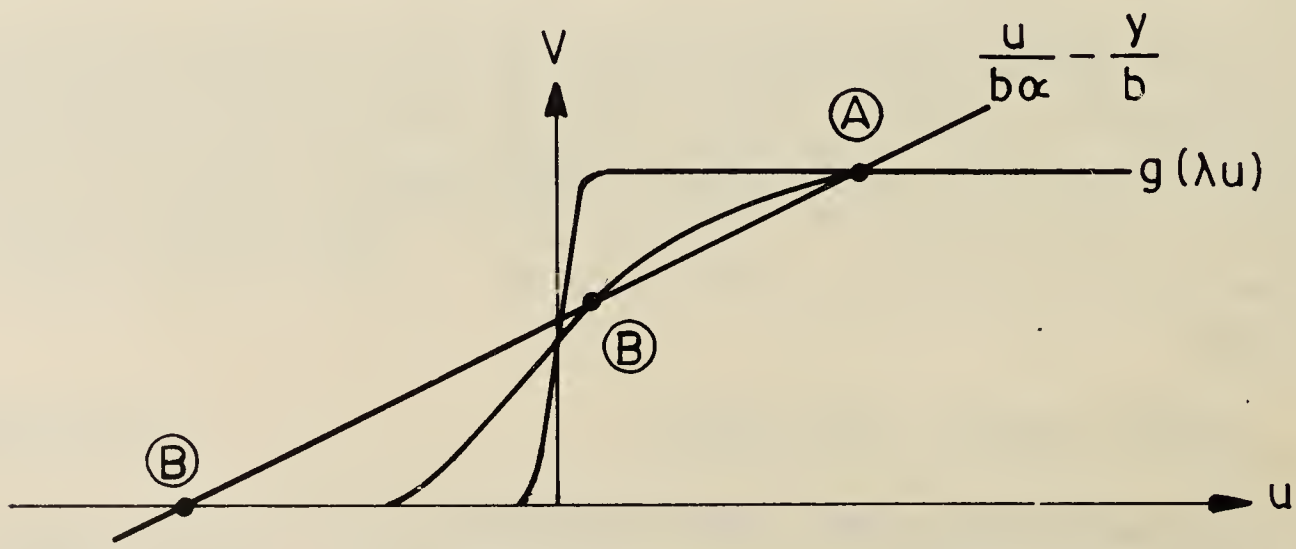


Figure 1. Equilibria in single-neuron case 1: B > 0.

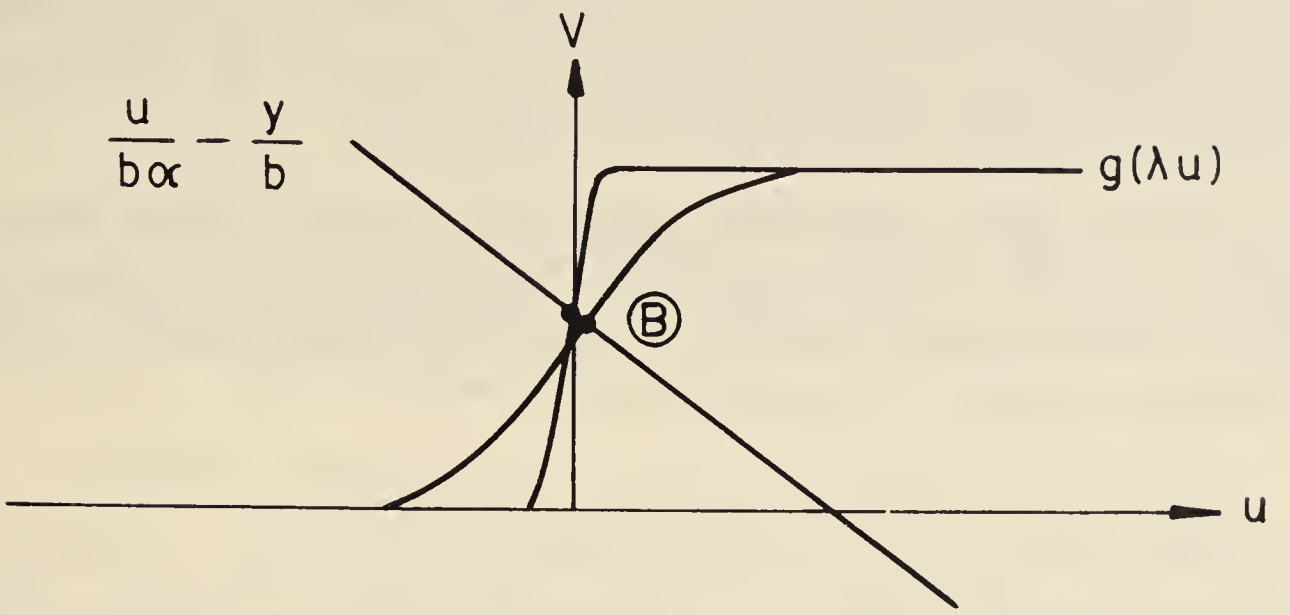


Figure 2. Equilibria in single-neuron case 2: B < 0.

4. Equilibrium in the interior of $\bar{H}$

To analyse the equilibria of the system (1), define

$$\alpha_i = R_i C_i,\ i = 1, \ldots, n;\ A = \operatorname{diag}\{\alpha_1, \ldots, \alpha_n\}, \tag{10}$$

$$b_{ij} = t_{ij}/C_i,\ y_i = I_i/C_i,\ i = 1, \ldots, n; \tag{11}$$

and define the map G: $R^n \to H$ by

$$[G(u)]_i = g_i(u_i). \tag{12}$$

Then the network equations (1) can be rewritten compactly as

$$\dot{u} = -A^{-1}u + BG(\lambda u) + y. \tag{13}$$

Now the equilibria of the system are at the solutions of

$$A^{-1}u = BG(\lambda u) + y. \tag{14}$$

In this section we are interested in equilibria in the interior of $\bar{H}$. If *all* components of V are to stay away from the limits 0 and 1 as $\lambda \to \infty$, then u must approach 0, while λu approaches some well-defined finite limit. Substituting $u = 0$ in (14) and noting that $V = G(\lambda u)$ gives

$$BV_{eq} + y = 0, \text{ or } V_{eq} = -B^{-1}y = -T^{-1}I. \tag{15}$$

Now $-T^{-1}I$ is the only *possible* equilibrium of (1) that can remain in the interior of $\bar{H}$ as $\lambda \to \infty$. Observe however that V_{eq} depends on the external voltage vector I and is thus a variable. Now we can state the main result of this section.

PROPOSITION 4.1.

Let S be a closed subset of the open hypercube H. Then there exists an $\varepsilon > 0$, *dependent on S, such that the following statements are true for every interconnection matrix T such that* $\| T - T_0 \| < \varepsilon$:

(1) *for all sufficiently large* λ, *the system* (1) *has at most one equilibrium inside S. As* $\lambda \to \infty$, *this equilibrium* (*if any*) *approaches* $-T^{-1}I$;

(2) *this equilibrium is hyperbolic for all sufficiently large* λ, *and the dimensions of its stable manifold and its unstable manifold* are the same as the number of negative and the number of positive eigenvalues of* T_0, *respectively.*

Proof. (1) It has already been shown that this statement is true *provided* T is nonsingular. But since T_0 is nonsingular, so is T provided ε is sufficiently small.

(2) To prove this statement, linearize the network equations around the equilibrium V_{eq} in V-space, or near $G^{-1}(V_{eq})$ in the u-space. Now

$$\frac{\mathrm{d}}{\mathrm{d}u}[-A^{-1}u + BG(\lambda u) + y] = -A^{-1} + \lambda BJ(\lambda u), \tag{16}$$

where J is the (diagonal) Jacobian matrix of the map G. As $\lambda \to \infty$, the quantity $\lambda u \to G^{-1}(-B^{-1}y)$, as shown earlier. Define

$$\bar{u} = G^{-1}(-B^{-1}y),\ \bar{J} = J(\bar{u}). \tag{17}$$

Then, as $\lambda \to \infty$, the term $\lambda B\bar{J}$ swamps the $-A^{-1}$ term. Therefore the behaviour of the solution trajectories near the equilibrium is determined by the eigenvalues of $B\bar{J}$.

To study these, let us begin with the simple case where $T = T_0$, i.e. the interconnection matrix equals the nominal interconnection matrix and is thus symmetric. It is a well-known result (see e.g. Gantmacher 1951, p. 297) that if M is any nonsingular matrix, then T_0 and $M^t T_0 M$ have the same *signature*, that is, the same number of positive, zero, and negative eigenvalues. Since T_0 is hyperbolic and therefore nonsingular by assumption, it follows that neither T_0 nor $M^t T_0 M$ has any zero eigenvalues. Now, since

$$T_0 M M^t = (M^t)^{-1}(M^t T_0 M)M^t, \tag{18}$$

it follows that $T_0 M M^t$ has the same eigenvalues as $M^t T_0 M$, and hence the same signature as T_0. Next, letting B_0 denote the matrix B obtained from (11) when $T = T_0$, we have

$$B_0 \bar{J} = C^{-1} T_0 \bar{J}, \tag{19}$$

where C is the diagonal matrix with C_i on the diagonal. Since C^{-1} and $\bar{J}$ are both diagonal, they commute. Consequently,

$$\bar{J}^{-1/2} C^{-1/2}[C^{-1/2}\bar{J}^{1/2} T_0 \bar{J}^{1/2} C^{-1/2}] C^{1/2}\bar{J}^{1/2} = C^{-1} T_0 \bar{J}. \tag{20}$$

Hence $B_0 = C^{-1} T_0 \bar{J}$ is similar to $C^{-1/2}\bar{J}^{1/2} T_0 \bar{J}^{1/2} C^{-1/2}$, which in turn has only real eigenvalues and has the same signature as T_0.

Now consider the case where T_0 is replaced by T, which is "near" T_0. Since T is *not* assumed to be symmetric, it is no longer true that $B_0\bar{J}$ has only real eigenvalues. To handle this case we proceed as follows: Given the set S in H, define

$$G^{-1}(S) = \{u \in R^n : G(u) \in S\}, \tag{21}$$

$$\mathbf{J} = \{J(u) :\ u \in G^{-1}(S)\}, \tag{22}$$

---

* See Hirsch & Smale (1974) for definitions of these terms.

and note that $\mathbf{J}$ is compact. By continuity, for each $J_0 \in \mathbf{J}$ there exist an $\varepsilon > 0$ and a $\delta > 0$ such that, whenever $\| T - T_0 \| < \varepsilon$ and $\| J - J_0 \| < \delta$, the matrices $C^{-1}TJ$ and $C^{-1}T_0J_0$ have the same number of eigenvalues with negative and positive real parts, which in turn is the same as the number of negative and positive eigenvalues of T_0. Of course ε and δ depend on J_0. Now let $B(J_0, \delta)$ denote the ball of radius δ centred at J_0, and let J_0 vary over S. Then the balls $B(J_0, \delta)$ cover S. Since S is compact, a finite subset of these balls is enough to cover S, say $B(J_1, \delta_1), \ldots, B(J_k, \delta_k)$. Pick the corresponding $\varepsilon_1, \ldots, \varepsilon_k$, and let

$$\varepsilon = \min\{\varepsilon_1, \ldots, \varepsilon_k\}. \tag{23}$$

Then, whenever $\| T - T_0 \| < \varepsilon$, we see that $C^{-1}TJ$ has the same number of eigenvalues with negative and positive eigenvalues as does T_0. This completes the proof of the second part.

Example 4.2. As an illustration of proposition 4.1, consider the A/D converter circuit of Tank & Hopfield (1986). If we study the four-bit converter, then $n = 4$, and

$$T = \begin{bmatrix} 0 & -2 & -4 & -8 \\ -2 & 0 & -8 & -16 \\ -4 & -8 & 0 & -32 \\ -8 & -16 & -32 & 0 \end{bmatrix}, \quad I = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 8 \end{bmatrix} x - \begin{bmatrix} 0{\cdot}5 \\ 2 \\ 8 \\ 32 \end{bmatrix}, \tag{24}$$

where x is the real number which is to be quantized. This neural network evolves on the four-dimensional open hypercube $H = (0, 1)^4$. The objective of the example is to determine the range of values of x for which the network has an equilibrium in the interior or H, and to determine the dimensions of its stable and unstable manifolds.

Taking the second question first, it is easy to verify that T_0 has two negative and two positive eigenvalues. Thus, if the network has an equilibrium in the interior of $\bar{H}$, it is hyperbolic, and its stable and unstable manifolds both have dimension two.

Next, we compute

$$V_{eq} = -\,T^{-1}I = \begin{bmatrix} -2 \\ -0{\cdot}75 \\ -0{\cdot}125 \\ -0{\cdot}1875 \end{bmatrix} + \begin{bmatrix} 1/3 \\ 1/6 \\ 1/12 \\ 1/24 \end{bmatrix} x. \tag{25}$$

It is routine to verify that the above vector belongs to the open hypercube H if and only if $6 < x < 9$. Thus the neural network corresponding to the four-bit A/D converter has an equilibrium in the interior of $\bar{H}$ if and only if x belongs to the open interval $(6, 9)$.

5. Equilibria in the corners

In this section, we study whether any equilibria of the system (1) approach the corners of the hyercube $\bar{H}$ as the sigmoid gain $\lambda \to \infty$. Recall that $\mathbf{b}$ denotes the Boolean set $\{0, 1\}$, so that $\mathbf{b}^n$ is the set of corners of the closed hypercube $\bar{H}$. Now, since the differential equation (13) evolves on the *open* hypercube H, no vector in $\mathbf{b}^n$ can actually be an equilibrium of this system. However, it is possible that, as $\lambda \to \infty$, some equilibria of (13) *approach* a vector in $\mathbf{b}^n$.

PROPOSITION 5.1.

Let e be an arbitrary vector in $\mathbf{b}^n$. *Then an equilibrium of* (13) *approaches e as* $\lambda \to \infty$ *if and only if e satisfies the parity condition, defined as follows: Let* $z = Te + I$. *Then*

$$z_i > 0 \text{ if } e_i = 1,\ z_i < 0 \text{ if } e_i = 0. \tag{26}$$

Proof. Put $\dot{u} = 0$ in (13). This gives

$$0 = -A^{-1}u + BG(\lambda u) + y, \tag{27}$$

or

$$u = A[BG(\lambda u) + y]. \tag{28}$$

Now, if we substitute $G(\lambda u) = V = e \in \mathbf{b}^n$, then we get

$$u_{eq} = A[Be + y] = AC^{-1}z. \tag{29}$$

Thus, as $\lambda \to \infty$, $V = G(\lambda u_{eq}) \to e$, *provided*

$$(u_{eq})_i > 0 \text{ if } e_i = 1, (u_{eq})_i < 0 \text{ if } e_i = 0. \tag{30}$$

But since $u_{eq} = AC^{-1}z$, $(u_{eq})_i = R_i z_i$ for all i, and it follows that each component of u_{eq} has the same sign as the corresponding component of z. Hence (30) is equivalent to (26).

PROPOSITION 5.2.

Suppose that an equilibrium of (13) *approaches an element of* $\mathbf{b}^n$ as $\lambda \to \infty$. *Then this equilibrium is exponentially stable for all sufficiently large* λ.

Proof. Linearize (13) around the equilibrium u_{eq} of (29). The Jacobian matrix of the right side of (13) at u_{eq} is

$$-A^{-1} + BJ(\lambda u_{eq})\lambda. \tag{31}$$

By assumption, $\lambda J(\lambda u_{eq}) \to 0$ as $\lambda \to \infty$. Hence the Jacobian approaches $-A^{-1}$, which is a Hurwitx matrix.

Remark. An informal, but informative, way to state the above proposition is: "All equilibria near the corners of $\bar{H}$ are asymptotically stable."

Example 5.3. Consider again the four-bit A/D converter of example 4.2. In Tank & Hopfield (1986) it is claimed that, if x is any real number and if the neural network is started from the zero initial state (i.e. $u_i = 0$ for all i), then eventually the vector V will converge to the correct binary quantization of the real number x. However, for a given x there could be more than one stable equilibrium of the neural network, and depending on the initial condition the solution trajectory of the neural network could converge to an incorrect binary vector. If x is not kept fixed but is changed periodically, then it is necessary to "re-initialize" the network each time x is changed. Otherwise the solution trajectory will converge to an incorrect value. This problem is referred to in Tank & Hopfield (1986) as "hysteresis."

Since the neural network has four neurons, there are $2^4 = 16$ possible binary vectors,

or 16 corners to the hypercube $\bar{H}$. By taking each corner in turn, it is possible to determine the values of x for which an equilibrium exists at that corner. This can be done using proposition 5.1. By proposition 5.2, each such equilibrium is asymptotically stable. Hence, for some initial values of u at least, the solution trajectory will converge to that corner.

To illustrate the application of proposition 5.1, consider the corner $e = [1\ 0\ 1\ 1]^t$. Note that the first component represents the lowest or least significant bit whereas the last component represents the highest bit. Hence this vector corresponds to the binary representation of the integer 13. To determine for what values of x an equilibrium exists near this corner, we compute the vector $Be + y$, as per proposition 5.1. This gives

$$Be + y = -\begin{bmatrix} 12{\cdot}5 \\ 28 \\ 44 \\ 56 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 4 \\ 8 \end{bmatrix} x. \tag{32}$$

Now, in order for an equilibrium to exist near this corner, a necessary and sufficient condition is that

$$-12{\cdot}5 + x > 0,\ -28 + 2x < 0,\ -44 + 4x > 0,\ -56 + 8x > 0. \tag{33}$$

Solving these inequalities shows that an equilibrium exists near this corner if and only if

$$12{\cdot}5 < x < 14. \tag{34}$$

The same process can be repeated at all sixteen binary vectors, and corresponding intervals of x can be computed. This is displayed in table 1. (It is easy to show, using proposition 5.1, that the set of values of x corresponding to a given binary vector is always an interval.) For ease of presentation, the sixteen binary vectors have been shown in terms of the corresponding decimal integer.

From the table one can see that, corresponding to a given real number x for which it is desired to find a binary quantization, there can be as many as *three* distinct asymptotically stable equilibria. Moreover, these equilibria need not be anywhere close to the correct binary quantization. For example, if $x = 4{\cdot}3$, then there are three asymptotically stable equilibria, at (in decimal representation) $e = 3, 4, 8$. As per the convention of Tank and Hopfield, if $3{\cdot}5 < x < 4{\cdot}5$, then the correct binary quantization is 4. Hence one would hope that the neural network would converge towards the corner $e = 4 = [0\ 0\ 1\ 0]^t$. But since there are two other asymptotically stable equilibria, for suitable initial conditions the neural network will in fact converge

Table 1. Range of equilibria for A/D converters.

e	x	e	x	e	x
0	$x < 0{\cdot}5$	1	$0{\cdot}5 < x < 2$	2	$1 < x < 2.5$
3	$2{\cdot}5 < x < 5$	4	$2 < x < 4{\cdot}5$	5	$4.5 < x < 6$
6	$5 < x < 6{\cdot}5$	7	$6{\cdot}5 < x < 9$	8	$4 < x < 8{\cdot}5$
9	$8{\cdot}5 < x < 10$	10	$9 < x < 10{\cdot}5$	11	$10{\cdot}5 < x < 13$
12	$10 < x < 12{\cdot}5$	13	$12{\cdot}5 < x < 14$	14	$13 < x < 14{\cdot}5$
15	$14{\cdot}5 < x$				

towards the corners $e = 3$ or 8. Now one can console oneself that the corner $e = 3$ is "only off by one" and call it hysteresis, but one would certainly not be willing to accept an answer of 8 when the desired answer is 4. This brings up the question of whether there is an improved version of an A/D converter which does not exhibit such multiple asymptotically stable equilibria. The answer is "yes", as shown in Vidyasagar (1991).

6. Equilibria in the faces of $\bar{H}$

So far we have studied the existence of equilibria in the interior of $\bar{H}$, and near the corners of $\bar{H}$. In this section, we complete the analysis by studying conditions under which there exist equilibria in the *faces* of $\bar{H}$, i.e. equilibria where some components approach 0 or 1 while the remaining components remain bounded away from 0 and 1 as $\lambda \to \infty$.

We are searching for solutions to

$$A^{-1}u = Be + y = z, \tag{35}$$

where *some components* of e belong to $\{0, 1\}$ while the remaining components belong to the open interval $(0, 1)$. Note that if some component of e belongs to $(0, 1)$, then the corresponding component of u_{eq} (and of z) must be zero; otherwise $g(\lambda u) \to 0$ or 1 as $\lambda \to \infty$.

To check for such a solution, suppose without loss of generality that the indices $\{1, \ldots, n\}$ have permuted such that the first k components of z are zero while the remaining $n - k$ are nonzero. Thus the first k components of e belong to $(0, 1)$ while the rest belong to $\{0, 1\}$. Define

$$e_a = [e_1 \ldots e_k]^t,\ e_b = [e_{k+1} \ldots e_n]^t, \tag{36}$$

and partition y, z, and B commensurately. Then, in partitioned form, (35) becomes

$$0_k = B_{aa}e_a + B_{ab}e_b + y_a, \tag{37}$$

$$z_b = B_{ba}e_a + B_{bb}e_b + y_b, \tag{38}$$

where $e_a \in H^k$ and $e_b \in \mathbf{b}^{n-k}$. Now (37) can be solved for e_a if B_{aa} is nonsingular. By assumption, all principal submatrices of T_0 of size 2×2 or larger are hyperbolic and thus nonsingular. Hence B_{aa} is nonsingular provided $k \geqslant 2$ and T is sufficiently close to T_0. However, if $k = 1$ then B_{aa} is just a scalar which will be zero if there are no self-interactions. In this case (37) reduces to

$$0 = B_{ab}e_b + y_a \tag{39}$$

Since $e_b \in \mathbf{b}^{n-1}$, it can assume one of only 2^{n-1} distinct (vector) values. Hence the set of y_a for which (39) can be satisfied is of type LV. This discussion can be summarized as follows:

PROPOSITION 6.1.

If the neural network has no self-interactions, then for all inputs except for those belonging to a set of type LV, there will be no equilibria which approach an edge of $\bar{H}$ as $\lambda \to \infty$.

Now back to (37)–(38) in the case where $k \geqslant 2$. In this case one can solve (37) to obtain

$$e_a = -B_{aa}^{-1}(B_{ab}e_b + y_a). \tag{40}$$

Now fixing $e_b \in \mathbf{b}^{n-k}$ and allowing e_a to vary over H^k defines a face of $\bar{H}$ of dimension k. Depending on e_b ad y_a, the e_a found from (40) may or may not belong to H^k. This is the first condition to be satisfied. The second condition is obtained by substituting for e_a in (38), which gives

$$z_b = (B_{bb} - B_{ba}B_{aa}^{-1}B_{ab})e_b + y_b - B_{ba}B_{aa}^{-1}y_a. \tag{41}$$

This vector has to satisfy the parity condition that $(z_b)_i > 0$ if $(e_b)_i = 1$ and $(z_b)_i < 0$ if $(e_b)_i = 0$. This discussion can be summarized as follows:

PROPOSITION 6.2.

Pick an $e_b \in \mathbf{b}^{n-k}$. *Then, as* $\lambda \to \infty$, *there is an equilibrium of* (13) *approaching the face of* $\bar{H}$ *defined by letting* e_a *vary over* H^k, if and only if two conditions are satisfied:

(i) e_a *as defined in* (40) *belongs to* H^k, *and*
(ii) *if* z_b *is defined by* (41), *then* z_b *satisfies the parity condition with respect to* e_b; *i.e.* $(z_b)_i > 0$ *if* $(e_b)_i = 1$ *and* $(z_b)_i < 0$ *if* $(e_b)_i = 0$.

In addition, for all inputs except those belonging to a set of type LV, there are only a finite number of equilibria in the faces of $\bar{H}$.

Example 6.3. (Three-bit A/D converter) As an illustration of proposition 6.2, consider the same Tank and Hopfield A/D converter circuit, but this time with only three neurons, so that it does a three-bit quantization of a given real number. In this case,

$$T = B = \begin{bmatrix} 0 & -2 & -4 \\ -2 & 0 & -8 \\ -4 & -8 & 0 \end{bmatrix}, \quad I = y = -\begin{bmatrix} 0{\cdot}5 \\ 2 \\ 4 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix} x. \tag{42}$$

Let $x = 3{\cdot}2$; we show that it is possible to obtain a complete characterization of all equilibria of the neural network.

First, compute

$$V_{eq} = -B^{-1}y = \begin{bmatrix} 0{\cdot}35 \\ 0{\cdot}425 \\ 0{\cdot}4625 \end{bmatrix} \tag{43}$$

Since $V_{eq} \in (0, 1)^3$, there is indeed an equilibrium at this point as $\lambda \to \infty$, i.e. as the neural characteristics approach those of an ideal switch. Next, let us check for equilibria in the corners of $\bar{H}^3$. Using the procedure of proposition 5.1 as illustrated in example 5.2, one finds that there are (asymptotically stable) equilibria only at $e = [0\ \ 0\ \ 1]^t = 4$ and at $e = [1\ \ 1\ \ 0]^t = 3$. Finally, let us check for solutions of (38) in the faces of $\bar{H} = [0, 1]^3$. First, since all diagonal elements of B are zero, it follows from proposition 6.1 that there are no equilibria along the edges of the cube $\bar{H}$. Next we try setting one component of e equal to zero and solving for the other two. If we

set $e_1 = 0$, then solving (37) gives

$$\begin{bmatrix} e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} 0{\cdot}6 \\ 0{\cdot}55 \end{bmatrix} \in (0,1)^2. \tag{44}$$

Thus it can be concluded that, as $\lambda \to \infty$, there will be an equilibrium near $V = [0 \;\; 0{\cdot}6 \;\; 0{\cdot}55]^t$. Similarly it can be verified that there will be another equilibrium near $[1 \;\; 0{\cdot}1 \;\; 0{\cdot}3]^t$, and that these are the only equilibria along the faces of $\bar{H}^3$.

Now let us study the stability of the equilibria in the faces. Once the integer k and the vector $e_b \in \mathbf{b}^{n-k}$ are fixed, proposition 6.2 allows one to determine whether there exists an equilibrium in the corresponding face of $\bar{H}$. To study the stability of this equilibrium, define

$$\Lambda_k = \begin{bmatrix} \lambda I_k & 0 \\ 0 & I_{n-k} \end{bmatrix}, \; u_k = \Lambda_k u. \tag{45}$$

Then from (13) it follows that

$$\dot{u}_k = \Lambda_k \dot{u} = -\Lambda_k A^{-1} u + \Lambda_k BG(\lambda u) + \Lambda_k y \tag{46}$$

$$= -A^{-1} u_k + \Lambda_k BG(\lambda \Lambda_k^{-1} u_k) + \Lambda_k y. \tag{47}$$

Here we have used the obvious fact that

$$\Lambda_k A^{-1} \Lambda_k^{-1} = A^{-1}, \tag{48}$$

since all matrices are diagonal. Now let $\lambda \to \infty$. Then the fact that the conditions of proposition 6.2 are satisfied ensures that

$$G(\lambda \Lambda_k^{-1} u_k) \to \begin{bmatrix} e_a \\ e_b \end{bmatrix} = e. \tag{49}$$

Define

$$u_k* = G^{-1}(e). \tag{50}$$

Now linearize (13) around the equilibrium in u_k-space. The Jacobian matrix is

$$-A^{-1} + \Lambda_k BJ(\lambda \Lambda_k^{-1} u_k*) \lambda \Lambda_k^{-1}. \tag{51}$$

Now consider separately the matrix

$$M = J(\lambda \Lambda_k^{-1} u_k*) \lambda \Lambda_k^{-1}. \tag{52}$$

This is a diagonal matrix; moreover

$$m_{ii} = g_i'[(u_k)_i], \text{ for } 1 \leqslant i \leqslant k, \tag{53}$$

$$m_{ii} = g_i'[\lambda u_i)] \to 0 \text{ as } \lambda \to \infty, \text{ for } k+1 \leqslant i \leqslant n. \tag{54}$$

Hence, as $\lambda \to \infty$,

$$M \to \begin{bmatrix} M_a & 0 \\ 0 & 0 \end{bmatrix}, \tag{55}$$

$$-A^{-1} + \Lambda_k BM \to \begin{bmatrix} -A_a^{-1} + \lambda B_{aa} M_a & 0 \\ B_{ba} M & -A_b^{-1} \end{bmatrix}. \tag{56}$$

As $\lambda \to \infty$, the matrix becomes block-triangular. Moreover, in the upper left block, the $-A_a^{-1}$ term becomes insignificant in comparison to the $\lambda B_{aa} M_a$ term. Thus the spectrum of the linearized system approaches

$$\operatorname{spec}(\lambda B_{aa} M_a) \cup \operatorname{spec}(-A_b^{-1}). \tag{57}$$

Of course the spectrum of $-A_b^{-1}$ is just $-\{\alpha_{k+1}^{-1}, \ldots, -\alpha_n^{-1}\}$. Now, as in the proof of proposition 4.1, it is easy to show that the matrix $B_{aa} M_a$ has the same number of eigenvalues with negative and positive real parts as does the matrix $(T_0)_{aa}$, provided $\| T - T_0 \|$ is sufficiently small. This discussion can be summarized as follows:

PROPOSITION 6.4.

Let $J = \{j_1, \ldots, j_k\}$ be a subset of $\{1, \ldots, n\}$, with $2 \leqslant k \leqslant n-1$. Pick a value of either 0 *or* 1 *for each $e_j, j \notin J$, and consider the corresponding k-dimensional face of $\bar{H}$ defined by*

$$e_j \in (0, 1) \text{ for } j \in J. \tag{58}$$

Let S be any closed subset of this face, and define the $k \times k$ principal submatrix of T_0:

$$(T_0)_{JJ} = [(T_0)_{ij}, i, j \in J]. \tag{59}$$

Suppose $(T_0)_{JJ}$ has ν positive eigenvalues. Then there exists an $\varepsilon > 0$ such that the following statements are true whenever $\| T - T_0 \| < \varepsilon$.

(i) As $\lambda \to \infty$, at the most one equilibrium of (13) approaches S, and it can be found using proposition 6.2.

(ii) This equilibrium is hyperbolic. Its stable and unstable manifolds have dimensions $n - \nu$ and ν respectively.

Remark. Once the index set J is fixed, there are 2^{n-k} different combinations of values that can be assigned to the components $e_j, j \notin J$. Proposition 6.4 makes it clear that equilibria in *each* of these faces, if any, have the same "signature." Put another way, equilibria in "opposite" faces of $\bar{H}$ have the same signature.

Example 6.5. Let us continue example 6.3. The analysis previously carried out shows that there is an equilibrium at $V_{eq} = [0{\cdot}35 \;\; 0{\cdot}425 \;\; 0{\cdot}4625]^t$. Now the matrix T of (42) has one negative and two positive eigenvalues. Accordingly, from proposition 4.1, this equilibrium has a stable manifold of dimension one and an unstable manifold of dimension two. Next, there are asymptotically stable equilibria at $e_1 = [0 \;\; 0 \;\; 1]^t$ and $e_2 = [1 \;\; 1 \;\; 0]^t$. Now consider equilibria in the faces. Letting $J = \{2, 3\}$ and assigning $e_1 = 0$ leads to the equilibrium at $V_1 = [0 \;\; 0{\cdot}6 \;\; 0{\cdot}55]^t$, whereas assigning $e_1 = 1$ leads to the equilibrium $V_2 = [1 \;\; 0{\cdot}1 \;\; 0{\cdot}3]^t$. These equilibria are in opposite faces of the three-dimensional cube $[0, 1]^3$. Now

$$(T_0)_{JJ} = \begin{bmatrix} t_{022} & t_{023} \\ t_{032} & t_{033} \end{bmatrix} = \begin{bmatrix} 0 & -8 \\ -8 & 0 \end{bmatrix}. \tag{60}$$

This matrix has one positive eigenvalue. This shows that both V_1 and V_2 have stable manifolds of dimension two and an unstable manifold of dimension one.

The most important point to note about this example is that *all the above conclusions remain valid even if the interconnection matrix is perturbed slightly from its original*

symmetric value. Of course the actual values of the various equilibria will change slightly in a continuous fashion, but the dimensions of the various stable and unstable manifolds will not change.

7. Rate of convergence of trajectories

In this section some preliminary results are given about the rate at which the equilibria of the system (13) approach the corners of $\bar{H}$, and the rate at which the solution trajectories approach these equilibria.

Suppose e is a vector in $\mathbf{b}^n$, i.e. suppose e is a corner point of the hypercube $\bar{H}$. Then proposition 5.1 states that the system (13) has an equilibrium approaching e if and only if the vector $z = Te + I$ has the same "parity" as e. Thus, in proposition 5.1, only the signs of the various components of z are pertinent, and their magnitudes do not play any role in determining whether or not *there exists* an equilibrium near a particular corner. Now it is shown that the magnitudes of the components of z do determine the *speed of convergence* of the equilibrium to e as the sigmoid gain $\lambda \to \infty$.

To be specific, suppose all the neural characteristics are identical, and are given by

$$g_i(x) = 1/(1 + e^{-x}), \quad \text{for } i = 1, \ldots, n. \tag{61}$$

Suppose $e \in \mathbf{b}^n$ and that $z = Te + I$ has the same parity as e. In accordance with (26), define

$$u_{eqi} = (\alpha_i/\beta_i) z_i = R_i z_i, \quad \text{for } i = 1, \ldots, n. \tag{62}$$

Now define

$$V_{eq} = G(\lambda u_{eq}), \tag{63}$$

and let $\lambda \to \infty$, i.e. let the sigmoid characteristics become steeper and steeper

PROPOSITION 7.1.

Let all symbols be as defined above. Then

$$\lim_{\lambda \to \infty} \frac{\ln|e_i - v_{eqi}|}{\ln|e_j - v_{eqj}|} = \frac{|u_{eqi}|}{|u_{eqj}|}. \tag{64}$$

Proof. Suppose first that $u_i > 0$, $e_i = 1$ (by the parity condition). Then, as $\lambda \to \infty$, we have

$$v_{eqi} = 1/[1 + \exp(-\lambda u_{eqi})] \approx 1 - \exp(-\lambda u_{eqi}), \tag{65}$$

$$\ln(1 - v_{eqi}) \approx -\lambda u_{eqi}. \tag{66}$$

Now suppose $u_i < 0$, $e_i = 0$. Then, as $\lambda \to \infty$, we have

$$\ln v_{eqi} = -\ln[1 + \exp(-\lambda u_{eqi})] \approx -\ln[\exp(-\lambda u_{eqi})] = \lambda u_{eqi}. \tag{67}$$

The relationship (64) now follows readily from (66) and (67).

Proposition 7.1 addresses the issue of the rapidity with which V_{eq} approaches the

corner e as $\lambda \to \infty$. Basically, the larger the value of $|u_{eqi}|$, the more rapidly v_{eqi} approaches e_i. One can also explore the time behaviour of the solution trajectories of (13) for a fixed "large" value of λ.

PROPOSITION 7.2.

Let all symbols be as defined above. Then

$$\lim_{\lambda \to \infty} \lim_{t \to \infty} \frac{\ln|v_i(t) - v_{eqi}|}{\ln|v_j(t) - v_{eqj}|} = \frac{\alpha_j}{\alpha_i}. \tag{68}$$

Proof. Suppose λ is "large" and that the initial condition $u_i(0)$ is "near" u_{eqi}. Then it follows from (31) that

$$u_i(t) \approx u_{eqi} + [u_i(0) - u_{eqi}]\exp(-t/\alpha_i). \tag{69}$$

Suppose $u_{eqi} > 0$. Then, in analogy with (66), we have

$$\begin{aligned} v_i(t) &= 1/\{1 + \exp[-\lambda u_i(t)]\} \approx 1 - \exp[-\lambda u_i(t)] \\ &\approx 1 - \exp\{\lambda[u_{eqi} + (u_i(0) - u_{eqi})e^{-t/\alpha_i}]\} \\ &= 1 - \exp(\lambda u_{eqi}) \cdot \exp[\lambda(u_i(0) - u_{eqi})e^{-t/\alpha_i}] \\ &\approx 1 - \exp(\lambda u_{eqi}) \cdot [1 + \lambda(u_i(0) - u_{eqi})e^{-t/\alpha_i}] \\ &\approx v_{eqi} - \lambda \exp(\lambda u_{eqi})[u_i(0) - u_{eqi}]\exp(-t/\alpha_i), \end{aligned} \tag{70}$$

$$\ln[|v_i(t) - v_{eqi}|] \approx -(t/\alpha_i) + \lambda u_{eqi} + \ln \lambda |u_i(0) - u_{eqi}|. \tag{71}$$

As $t \to \infty$ for a fixed λ, the first term on the right side dominates the rest. A similar approximation applies when $u_{eqi} < 0$. The desired result (68) now follows readily.

Proposition 7.2 shows that, in the case where all neurons are identical (i.e. $\alpha_i = \alpha$ for all i), the trajectory in the V-space converges to the equilibrium at essentially the same rate in all components.

Example 7.3. Consider again the four-bit A/D converter of examples 4.2 and 5.3. Suppose the input x equals, say, 5.4. In this case, from table 1, one sees that there are three equilibria, namely at $e = [1\ 0\ 1\ 0]^t = 5$, $[0\ 1\ 1\ 0]^t = 6$, and $[0\ 0\ 0\ 1]^t = 8$. Table 2 shows the corresponding values of $z = Te + I$.

From the table one can see that, in two out of the three cases (in fact the two which represent the best digital approximations to the given input x, the components of z are smallest in magnitude corresponding to the least significant bits, and largest in magnitude corresponding to the most significant bits. Thus, as the sigmoid nonlinearities

Table 2. Rate of convergence for A/D converters.

e	z
$[1\ 0\ 1\ 0]^t$	$[\ 0{\cdot}9\ -1{\cdot}2\ \ 9{\cdot}6\ \ 27{\cdot}2]^t$
$[0\ 1\ 1\ 0]^t$	$[-1{\cdot}1\ \ 0{\cdot}8\ \ 5{\cdot}6\ -36{\cdot}8]^t$
$[0\ 0\ 0\ 1]^t$	$[-3{\cdot}1\ -7{\cdot}2\ -18{\cdot}4\ \ 11{\cdot}2]^t$

become steeper and steeper ($\lambda \to \infty$), one would expect that the most significant bits converge meet rapidly to the "correct" values. The same phenomenon can be observed for almost all values of the input variable x. The details are routine and are left to the reader.

8. Existence of equilibria in the corners

Proposition 5.1 states that *if* the system (13) has any equilibria near the corners of $\bar{H}$, then these are asymptotically stable. But, under certain circumstances, there might be *no* equilibria near the corners of $\bar{H}$.

First a positive result.

PROPOSITION 8.1.

Suppose the interconnection matrix T satisfies the following conditions: (i) *T is symmetric, and all of its diagonal elements are zero;* (ii) *Every principal submatrix of T of size 2×2 or larger, including T itself, is hyperbolic and has at least one positive eigenvalue. Under these conditions, for all inputs I expect those belonging to a set of type LV, there exists at least one binary vector $e \in \mathbf{b}^n$ such that $Te + I$ has the same parity as e.*

Proof. The assumptions ensure that the neural network exhibits total stability, i.e. every solution trajectory converges to an equilibrium (Hirsch 1987). Propositions 4.1, 6.1 and 6.2 show that there can be no asymptotically stable equilibria except near the corners of $\bar{H}$, while proposition 6.2 guarantees that there can only be a finite number of equilibria in the faces of $\bar{H}$. All these facts plus total stability lead one to conclude that there must exist at least one asymptotically stable equilibrium near a corner of $\bar{H}$. By proposition 5.1, this is equivalent to the parity condition being satisfied at some corner of $\bar{H}$. This is the desired conclusion.

Now an example to show that proposition 8.1 is *not* valid if the interconnection matrix T is perturbed.

Example 8.2. Consider a two-neuron network with the interconnection matrix

$$T = \begin{bmatrix} -\varepsilon & -1 \\ -1 & \varepsilon \end{bmatrix}. \tag{72}$$

Then, by applying proposition 5.1, one can verify that if

$$0 < i_1 < \varepsilon, i_2 < -\varepsilon, \tag{73}$$

then *none of the four vectors* in $\mathbf{b}^2$ satisfies the parity condition. But by applying proposition 6.2, one can see that there is an equilibrium near

$$e_1 = (i_1/\varepsilon), e_2 = 0. \tag{74}$$

To determine the signature of this equilibrium, let

$$u_1 = g_1^{-1}(e_1), m_{11} = g_1'(u_1) > 0. \tag{75}$$

Then, by (57) the spectrum of the linearized system around the equilibrium is

asymptotically equal to

$$\{-\lambda m_{11}, -\alpha_2\}. \tag{76}$$

Hence this equilibrium is asymptotically stable.

The point of proposition 8.1 and example 8.2 is the following: Under ideal conditions, there is (almost) always an asymptotically stable equilibrium near a corner of $\bar{H}$. Since the parity condition of proposition 5.1 is just an algebraic relationship, it is easy to see that, *for each fixed input vector I*, there is a small allowed perturbation of T such that there continues to exist an equilibrium near some corner of $\bar{H}$. But example 8.2 shows that the order of the quantifiers cannot be interchanged: It is *not* true that there exists a small allowed perturbation of T for which there continues to exist an equilibrium near some corner of $\bar{H}$.

As a final comment, observe that the proof of proposition 8.1 is quite round-about and unsatisfactory. The parity condition involves only linear algebra, and as such one would expect to be able to find a proof of the proposition based purely on linear algebra.

9. Conclusions

In this paper, we have given a complete analysis of the equilibria of neural networks with linear interconnections, in the case where the interconnection matrix is "nearly" symmetric, but *without* assuming that the interconnection matrix is actually symmetric. This allows one to analyse, among other things, the effects of imperfect implementation on the behaviour of Hopfield neural networks.

If a neural network has symmetric interconnections, then (Hirsch 1987) the network exhibits *total* stability, i.e. all solutions approach an equilibrium. This means, for example, that there are no limit cycles, i.e. there are no nontrivial periodic solutions. This conclusion depends heavily on the ability to construct a total Lyapunov or energy function, and the energy function of Hirsch (1987) is only valid if the interconnection matrix is symmetric. Thus it is still an open question as to whether a network with "nearly" symmetric interconnections can exhibit limit cycles, and if so, under what conditions.

Another issue which is as yet unresolved, even in the symmetric interconnections case, is that of calculating (or at least estimating) the basin or domain of attraction of each asymptotically stable equilibrium, which we now know can only lie in the corners of the hypercube $\bar{H}$ if the interconnection matrix has zero diagonal elements. This is a topic for further research.

The author thanks Professor Bosco Leung of the University of Waterloo for drawing his attention to this problem, and Professor Hans Schneider of the University of Wisconsin for some valuable help.

References

Gantmacher R 1959 *Matrix theory* (New York: Chelsea) vol. 1

Hopfield J J 1982 Neural networks and physical systems with emergent collective computational capabilities. *Proc. Natl. Acad. Sci. (USA)* 79: 2554–2558

Hopfield J J 1984 Neurons with graded response have collective computational capabilities like those of two-state neurons. *Proc. Natl. Acad. Sci. (USA)* 81: 3088–3092

Hopfield J J, Tank D W 1985 'Neural' computation of decision optimization problems. *Biol. Cybernet.* 52: 141–152

Hirsch M W 1987 Convergence in neural nets. *Proc. Int. Joint Conf. on Neural Networks* (New York: IEEE) vol. 2, pp. 115–125

Hirsch M W, Smale S 1974 *Differential equations, dynamical systems and linear algebra* (New York: Academic Press)

Tank D W, Hopfield J J 1986 Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit. *IEEE Trans. Circ. Thy.* CAS-33: 533–541

Vidyasagar M 1991 Improved neural networks for analog to digital conversion. *Circuits, Syst. Signal Process.* 10: (to appear)

Stochastic analysis of versatile workcentres

R RAM and N VISWANADHAM

Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560012, India

Abstract. We analyse the system consisting of a highly capable workcentre, which processes a variety of part types, using queueing models. The various part types produced by the system have distinct arrival and processing durations that are stochastic in nature. When an arriving workpiece finds the machine busy, it waits in a pre-process storage buffer (queue); this buffer may be common for all the part types, or may be dedicated for that part type. Further, this buffer may be capable of holding only a finite number of workpieces, or may be of infinite capacity. When the machine changes over from producing one type of part to another, a setup operation of stochastic duration is necessary to adjust the machine and load the necessary tools for production of the next part type. This model is representative of a typical machining centre in an Automated Manufacturing System. We focus on GI/G/1 models and multiqueue polling models, and their variants. The important performance measures of the system obtained by queueing analysis are the part-type-wise values of the mean lead time, mean inventory level, and the mean machine utilisation.

Keywords. Stochastic analysis; versatile workcentres; automated manufacturing system; lead time; inventory level; machine utilisation.

1. Introduction

In this paper, we present stochastic models for flexible machining centres which are the basic processing nodes of an Automated Manufacturing System (AMS). The major goal of automated manufacturing, viz., high quality, low volume production of a variety of part types concurrently, with low lead times, is sought to be achieved, based on the capability of the individual workcentres to process different part types. Hence, an understanding of the operation of these workcentres in the face of demands to produce multiple varieties of parts is very important in a performance evaluation study of the automated manufacturing system.

The most important performance measures obtained from a study of this basic system are the *mean lead time* (the time duration from the entry of raw workpiece to the system to the completion of the processing operation on the workcentre), the *mean inventory level* (the mean number of workpieces waiting ahead of a newly arriving workpiece), and the *mean machine utilisation* for each of the part types produced. These performance measures are of great importance in manufacturing, especially so

in automated manufacturing where we seek to minimise the lead times and inventories (ideally, lead time must be just the processing time, and inventory level must be just the workpieces under processing), these performance measures are particularly important. In the terminology of the queueing model approach of our paper, these performance parameters correspond to the mean response time, the mean queue length and mean server utilisation respectively; the different part types are represented by multiple job classes.

1.1 *The model*

We solve the following system: there is a versatile machining centre capable of processing a variety of parts, of which there are N different types. Workpieces of part type i arrive to the system according to a general, independent and identically distributed interarrival process of mean *rate* λ_i. If the machine is unable to serve this arriving workpiece immediately, the workpiece joins a queue to wait for the machine; the queue may be dedicated to that particular part type, or may be common to all part types. Further, the queue may be of finite capacity (capable of accommodating a maximum of a finite number of workpieces, say K_i for part type i). For processing workpieces of part type i, the machine has to be setup for this part type; this requires a setup operation, which has a general independent and identically distributed (i.i.d.) duration represented by the random variable R_i. If this setup is not disturbed, subsequent waiting workpieces of part type i can be processed without incurring additional setup. The machining of a type i workpiece takes a duration which is general i.i.d. and is represented by B_i. Processed parts are assumed to leave the system immediately. Let T_i denote the lead time (system response time) and L_i the mean inventory level (queue length) for part type i; we shall omit the subscript i when we deal with a single part type. The various aspects of the system considered in this paper may be modelled to varying degrees of detail; the issues considered, and their simplest and highest degrees of detail are summarised in table 1.

To complete the description of the model, we need to specify how the machine chooses a particular part type (*scheduling policy*), and, once a particular part type is selected, how the waiting workpieces of that particular part type are processed (*service policy*). We shall refer to the combination of scheduling and service policies as *operation policy*. It must be noted that for certain operation policies, like the first-come-first-served and Bernoulli scheduling policies, the scheduling and service policies overlap; in such cases, for simplicity, we shall continue to refer to them as scheduling policies.

The scheduling policy determines which of the N different part types is taken up

Table 1. Aspects of the systems considered.

Issue	Simplest assumption	Most detailed assumption
Number of queues	Common queue for all parts	Separate queue for each part
Queue capacity	Infinite capacity	Finite capacity (for each part type)
Setup time	Treated as a part of processing time. (Not considered explicitly)	Distinct setup time for each part type. (Explicitly considered)

for processing next when the processing of a particular part type is completed. Examples of scheduling policy that can be clearly demarcated from service policies are:

Cyclic scheduling: We assume that there is a separate queue (which may be of finite capacity or infinite capacity) for each part type. The different part types are taken up in the cyclic order $1, 2, 3, \ldots, N-1, N, 1, 2$, etc. The processing of each part type is preceded by a corresponding setup operation for that part type. This type of scheduling has been widely discussed in the literature on the analysis of computer systems and computer communication networks and is referred to as *cyclic service or polling*. The two extensive survey articles by Takagi (1988, 1990) provide a comprehensive discussion on the modelling, analysis and applications of these and related systems.

Probabilistic scheduling: We assume separate queues for different part types. Whenever the machine finishes processing a part type, the part type i is chosen next with a probability $p_i, (0 < p_i < 1$ for $1 \leqslant i \leqslant N$, and $\Sigma_{i=1}^{N} p_i = 1)$. Each service is preceded by an appropriate setup. These models are referred to as *random polling* models; an analysis of the discrete time version of this system is developed by Kleinrock & Levy (1988).

Markovian scheduling: This is an extension of probabilistic scheduling. Each part type has a separate queue. When the machine completes processing for (say) type i parts, we next take up type j parts for machining with probability $p_{i,j}$, where $0 \leqslant p_{i,j} \leqslant 1$, for $1 \leqslant i, j \leqslant N$, and for each part i, $\Sigma_{j=1}^{N} p_{i,j} = 1$. We assume that the $N \times N$ matrix $[p_{i,j}]$ forms the transition matrix of a irreducible, discrete time Markov chain.

The service policies can be clearly distingushed from the scheduling policy for the cases mentioned above. The service policy determines how much service is carried out for each part type, once the machine is setup for it. The service policies widely considered in the literature are:

Exhaustive: Once the workcentre is setup for a part, say type i, machining of type i parts is continued till no further workpieces of type i are waiting. In particular, those workpieces that may arrive during the current setup are also machined during this setup.

Gated: The machine processes all and only those workpieces that are waiting when the setup is completed. In particular, those workpieces of this part type that may arrive during the current setup have to wait for the next setup to be processed.

Limited: The amount of machining carried out on a part type is limited by specifying an upper limit on either the number of workpieces that may be processed during a setup or the maximum time for which waiting workpieces of the part type may be taken up for processing. Accordingly, we have either

— *K-limited service*, where a maximum number, say L_i of a part type i workpieces may be processed during the current setup, or
— *T-limited service*, where processing of type i parts is allowed for a maximum duration L_{T_i}.

Further, parts may be taken up in either exhaustive or gated mode for processing

under each of the above service disciplines, giving us a total of four variants of limited service. The K-limited, exhaustive discipline is the one commonly treated in the literature.

Given one of the above mentioned scheduling policies, if all the N part types adapt the same service discipline, e.g., all the part types use the gated service policy, then we have a system with *homogenous* service policy. On the other hand, if different part types have distinct service policies, e.g., part 1 has exhaustive service, part 2 has gated service, part 3 has K-limited-gated service and so on, we have a system with *mixed* service.

The following are examples of operation policies where the scheduling and service policies overlap.

First come first served (*FCFS*): This assumption is commonly made in the literature whenever there are random arrivals: all the arriving workpieces wait in a single, infinite capacity queue, from which they are taken up in an FCFS fashion for processing. The GI/G/1 queueing models are based on this assumption. Such models have been extensively applied in the modelling of manufacturing systems as a network of queues, each queue representing a workcentre.

Bernoulli scheduling: This is an example of a non-exhaustive service discipline applicable to multiqueue systems (i.e., a separate queue for each part type). At each instant of completion of machining of part type i, if the queue of part i is not empty, the machine makes a random decision: with probability p_i, $(0 \leqslant p_i \leqslant 1)$, it processes

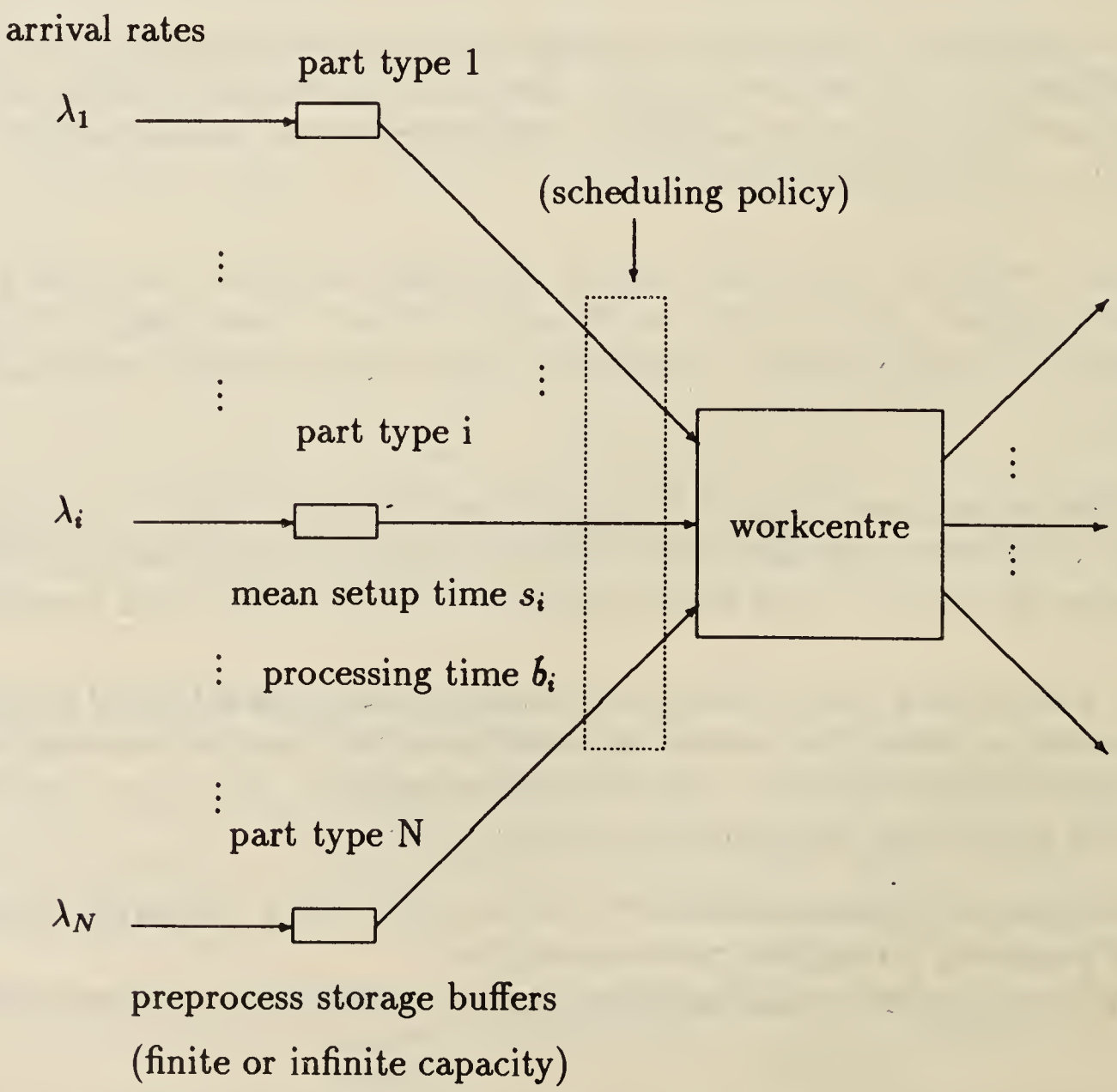


Figure 1. The basic model of a versatile workcentre.

the next waiting workpiece of the same part type, and with probability $1 - p_i$ it changes over to the next part type. If the queue of part i was empty at the completion of a type i workpiece machining, it changes over to the next part type with probability 1. When the machine switches from one part type to the next, a setup is carried out for the new part type before actual machining. Recently, Tedijanto (1990) presented an analysis of this service discipline; the symmetric version of this case has been exactly solved for the mean waiting time of each part type.

The basic model of the workcentre treated in this paper is depicted in figure 1.

The organisation of the paper is as follows: in the rest of this section, we present a brief survey of the literature on work related to this paper. In the next section, we present single queue, single server systems which are simplifications of the $GI_1, GI_2, \ldots, GI_N/G_1, G_2 \ldots, G_N/1/FCFS$ systems. Section 3 focusses on queueing analysis of multi-queue models, particularly cyclic server models. In § 4, we present numerical examples from the manufacturing context based on the models presented in this paper. Section 5 concludes the paper.

1.2 *Related literature*

The issue of scheduling multiple part varieties on a single machine under *known* demands, processing times and setup times (i.e., *the deterministic scheduling problem for a single machine*) has been well researched (Dobson *et al* 1987 and references therein). Several heuristic algorithms have been proposed for the control of this system under time varying stochastic demands, but assuming fixed production rates, e.g., Leachman & Gascon (1988). Recent investigations on the scheduling of manufacturing systems have been carried out using a hierarchical approach; in particular, the work of Gershwin (1989), Perkins & Kumar (1989), and Kumar & Seidman (1990) are particularly relevant. The hierarchical approach is based on a *time scaling* in the activities occurring in a manufacturing system: a part processing may take about an hour; a setup operation may take a duration an order of magnitude longer, say 10 hours; a machine failure may occur once in 200 hours. In these hierarchical production scheduling systems, the processing operations carried out by a workcentre are at the lowest level of the time scaling (fastest activities), and the setup activities are at the immediately higher level. The queueing models in this paper are utilised to analyse the system consisting of a versatile workcentre, along with its preprocess buffers, at these two lower time scales. However, in automated manufacturing an important aspect is the reduction of setup times through the use of mechanisms like automatic tool changers, and hence setup and operation times are typically of comparable magnitude. We have basically drawn upon the relevant results in the vast literature on single server queues the GI/G/1 queue, and its variants, and from the literature on *polling models* (Takagi 1988, 1990) developed particularly in the context of computer communication. Relevant literature on these models are surveyed and referenced at the appropriate point in the next two sections.

2. Single queue models

We start with queueing models where the different part types share a single common queue, which may be of infinite or finite capacity (see figure 2).

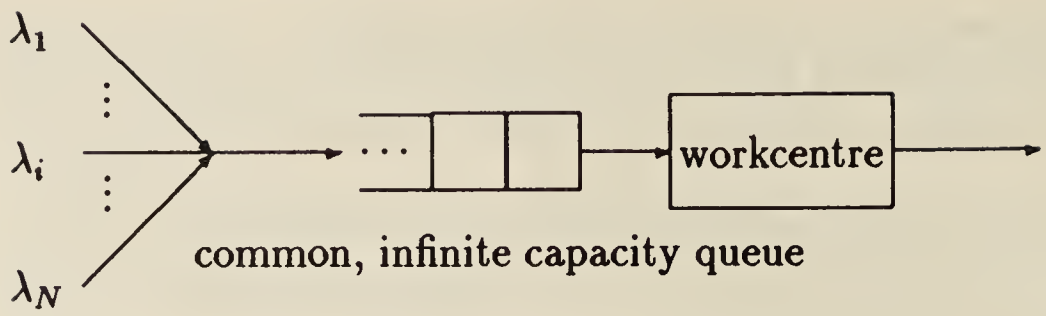


Figure 2. The common queue model.

2.1 *M/G/1 queueing model*

The simplest model of the system is the well known M/G/1 queue, widely described in the queueing literature. This entails that we make the following simplifying assumptions on the model:

(i) All arriving workpieces share a common, infinite capacity queue in an FCFS fashion.

(ii) The processing time is interpreted as the sum of the setup time and the actual machining time (i.e., the setup time is not considered explicitly). If B_i' represents the modified processing time of a workpiece of part type i, then

$$B_i' = B_i + p_{\text{setup},i} R_i,$$

where $p_{\text{setup},i}$ is the probability that a setup operation has to be carried out afresh. Since we have assumed exponential arrivals, $p_{\text{setup},i}$ is the same as the probability that the previous part type processed was *not* of type i, so $p_{\text{setup},i} = 1 - (\lambda_i / \Sigma_{j=1}^{N} \lambda_j)$. In terms of Laplace transforms, we have $\hat{B}_i'(s) = \hat{B}_i(s) \times [p_{\text{setup},i} \hat{R}_i(s) + 1 - p_{\text{setup},i}]$.

(iii) The total arrival rate to the system is $\lambda = \Sigma_{i=1}^{N} \lambda_i$, and the service time of an arbitrary workpiece has the transform

$$\hat{B}'(s) = \Sigma_{i=1}^{N} (\lambda_i / \lambda) \hat{B}_I'(s)$$

We are in effect analysing the $M_1,\ldots, M_N/G_1,\ldots, G_N/1$ queue by an M/G/1 model. The analysis of the system now proceeds along the standard method for the M/G/1 queue, which may be found in any standard text on queueing theory or stochastic modelling. A simple variant of this system, where each part type has distinct exponential arrival and service rates, i.e., the $M_1,\ldots, M_N/M_1,\ldots, M_N/1$ system, is discussed in an early paper by Ancker & Gafarian (1961).

2.2 *GI/G/1 queueing model*

2.2a *Related work*: Several investigations have been carried out on the approximate analysis of the GI/G/1 queueing system, which constitutes a general model for the single machine multiple part type processing system; see Shanthikumar & Buzacott (1980) for a survey of important results, and recommended methods applicable for specific parameter ranges. This is also useful in approximate analysis of non-product-form open general queueing networks of automated manufacturing systems, by adapting the product-form networks idea of decomposing the network into independent nodes corresponding to machining centres. Approximate analysis of general open queueing networks, and further refinements, have been proposed based on this independence concept by several authors – Kuehn (1979), Marie (1979), Whitt (1983), Shanthikumar & Buzacott (1981, 1985) and Bitran & Tirupati (1988, 1989). In

particular, the investigations of Shanthikumar and Buzacott, and Bitran and Tirupati are directed towards general open queueing network models of manufacturing systems, the former dealing with single product networks, and the latter with multi-product networks. The core of all these investigations is an appropriate analysis of individual workcentres processing multiple types of parts, and different processing times for different parts, by a GI/G/1 model under a multi-product assumption. In fact, several of the approximate models for this system have been developed especially for application to the analysis of networks of GI/G/1 systems with arbitrary routing. Shanthikumar & Gocmen (1983) have applied the principle of decomposition of a queueing network into individual independent nodes in developing a heuristic analysis of a closed network of GI/G/1 queues. In this subsection, we summarise the recent approaches to solving this model in the manufacturing context.

2.2b *Simplifying assumptions:* The analyses in these investigations are carried out using the knowledge of the first two moments of the interarrival and service processes (the '*parametric analysis*' of general queueing networks). In the context of the versatile machine-multiple part types case, the application of the GI/G/1 queueing model entails the following assumptions:

- the setup time is not explicitly considered; it may be treated as part of the machining time.
- the arrival and service processes, which can be different for different part types, have to be suitably combined. In the approach of the earlier papers, the system performance measures are obtained for a 'typical' part type, which is representative of all the part types; in a recently developed alternative approach (particularly Bitran & Tirupati 1988, 1990), performance measures are obtained for each part type individually by simplifying the system into a two-part type system: one part type is the particular part type of interest, and the second is an aggregate part type representative of all the other part types.
- the arrivals wait in an infinite capacity queue common to all part types, from which workpieces are taken up for machining on a first-come-first-served basis.

2.2c *Approximations based on the aggregation of all part types into a single 'typical' part type:* Let λ_j and c_{aj}^2 denote respectively the mean *rate* and the squared coefficient of variation (s.c.v.) of the interarrival process for part type $j (1 \leqslant j \leqslant N)$, and τ_j and c_{sj}^2 denote the mean processing *time* and the squared coefficient of variation of the service process respectively. We have to approximate the system at two stages to apply the model: (i) the parameters corresponding to the various part types have to be combined into those corresponding to a 'typical' part type; (ii) depending on the composite parameters obtained for the 'typical' part type, a suitable approximation of the GI/G/1 queue is to be applied.

(i) *Arrival process of the typical part type* – The mean total arrival rate of the 'typical' part to the system is given by the sum of the arrival rates of individual parts: $\lambda = \Sigma_{j=1}^{N} \lambda_j$.

The s.c.v. of the arrival process is obtained by the hybrid approximation of Albin used by Whitt (1983) as

$$c_A^2 = w \sum_{j=1}^{N} (\lambda_j/\lambda) c_{aj}^2 + 1 - w,$$

where w is a weight, given by

$$w = [1 + 4(1 - \rho)^2(N - 1)]^{-1}.$$

(Here, ρ denotes the offered load, or the utilisation, of the machine, and is given by $\rho = \lambda\tau$.)

(ii) *The service process of the typical part* – The mean service time of the typical workpiece is given by $\tau = (\Sigma_{j=1}^{N} \lambda_j \tau_j)/\lambda$, and its squared coefficient of variation of the service process is obtained as (Whitt 1983)

$$\tau^2(c_S^2 + 1) = \left[\sum_{j=1}^{N} \lambda_j \tau_j^2 (c_{sj}^2 + 1)\right] \Big/ \left[\sum_{j=1}^{N} \lambda_j\right].$$

Depending on the values obtained for c_S^2 and c_A^2, we can adopt the methods suggested by Shanthikumar & Buzacott (1980) and Whitt (1983) for the approximate analysis of the GI/G/1 system. We omit the mathematical details of these computations. See Whitt (1983) for a further discussion on approximate computation of distribution of the waiting time. In all these cases, the mean system lead time (waiting time in queue plus processing time) for a 'typical' part type is obtained from Little's Law as $E[R] = E[L]/\lambda$.

The work of Shanthikumar & Buzacott (1980, 1981) and Whitt (1983) is based on representing a multiproduct, general open queueing network in terms of a single representative part type typical of all the part types; the service and arrival parameters at each node are modified to aggregate the behaviour of all the part types by this typical part type. The routing information of individual parts are usually deterministic; the aggregation methodology converts this into Markovian routing. (The deterministic routing of different part types is retained in an exact analysis if we model the system by special classes of queueing networks, like open product-form queueing network models (Baskett *et al* 1975) or the closely related quasi-reversible networks (Kelly 1979); in this paper we are dealing with more general queueing models of individual machines (nodes) a network of which does not fall in these exactly solvable categories.)

2.2d *Approximate analyses explicitly considering distinct part types*: Bitran & Tirupati (1988) point out that the randomisation of deterministic routing of the various part types to Markovian routing in the aggregation process can lead to significant errors in the computation of system performance measures. They propose a refinement for computing the squared coefficient of variation of the departure process of a specific part type, say i, from a node (which is a GI/G/1 queue). To compute the s.c.v. of the departure process of part type i from a particular workcentre, we consider the following simplified view of each workcentre: consider the specific part type i, and an aggregate part type, which represents all part types other than i. The s.c.v. $c_D^2(i)$ of the departure process of part i from the workcentre may be computed from

$$c_D^2(i) = [(\lambda_i/\lambda)c_D^2] + c_{n(i)}^2,$$

where the new term $n(i) = z(i) + 1$, where $z(i)$ is the number of workpieces of the aggregate type that arrive during an interarrival time of part type i, $c_{n(i)}^2$ represents the s.c.v. of $n(i)$. The difficult part is estimating the s.c.v. $c_{n(i)}^2$. The authors propose three approximations for this purpose, which may be briefly stated as follows: (i) the arrivals of the aggregate product may be treated as Poisson; (ii) during any interarrival

period of the part type i (aggregate part type), the interarrival process of the aggregate part type (part type i) may be treated as Erlangian; and (iii) the arrival of part type i is treated as a random incidence in the arrival stream of the aggregate product; both the part type i and the aggregate part type have Erlang arrivals. Extensive numerical investigations by Bitran & Tirupati (1988) indicate that the proposed approximations provide significant improvements in accuracy over the aggregation approach previously employed. In a subsequent paper (Bitran & Tirupati 1989), they develop an approximate analysis for a single workcentre multiple-item system, where the arriving workpieces of the different part types have to form a batch of a given (fixed) size before service can be started.

In a related paper, Whitt (1988) has developed the theory for the output process for a particular part type from a node, when that part type is in light traffic. The basic idea of the light traffic approximation for the departure process of a single part type from a node may be stated thus: "If the arrival rate of one class (part type) to some queue (machine) is a small proportion of the total arrival rate there, then the departure process for that class from that queue tends to be nearly the same as the arrival process for that class to that queue". This principle can be used in an approximate analysis of a general multiclass queueing network. A simplified characterisation of the departure process of a GI/G/1 system with multiple arrival streams (part types) by a renewal process is developed by Albin (1986).

2.3 *Finite capacity queues*

Thus far we have not put any restriction on the capacity of the preprocess buffer queue. In real life manufacturing, the number of fixtures available is limited, and so is the space available to hold waiting workpieces in front of a machine. Hence, it is more realistic to assume a finite capacity for the preprocess queue. In the following, we summarise the results for finite buffer systems.

2.3a M/G/1/N *queue model*: This is finite capacity analog of the M/G/1 model. We may think of the well known M/M/1/N system as an elementary version of this model, where all the part types are aggregated (logically) into one representative part type, which has exponential arrival and service in a single server system with a total buffer capacity of N, including the server. Basharin (1965) has analysed the finite queue analog of the case where each part type has distinct exponential arrival and service in a finite capacity queue, single server system, i.e., the $M_1,\ldots. M_N/M_1,\ldots,M_N/1/N$. The general M/G/1/N system is analysed by Lavenberg (1975). An arrival that finds all the N buffer spaces full is assumed to be lost. Lavenberg presents an expression for the Laplace–Stieltjes Transform (LST) of the distribution of the queueing time in the system, in terms of the steady state probability of the imbedded Markov chain at the departure epochs, and the LST of the service time distribution. The relevant mathematical results are highly detailed, hence we omit them.

2.3b *The GI/G/1/N model*: This is the finite queue analog of the previous model, representing an important real life issue, viz., the availability of only a finite number of buffer spaces to hold waiting part types in front of a workcentre. An analysis of this system will also be fundamental to an analysis of a network of finite capacity queues, i.e., general open queueing networks with blocking.

3. Multiqueue models

We next turn our attention to queueing models, where each part type forms a separate queue, which are attended to by the versatile workcentre represented by a single server. Typically, the different part types are served in cyclic order, and each queue is assumed to have infinite capacity. Few results are available for the finite-capacity queues case.

3.1 *Cyclic server (polling) models with infinite buffer queues*

This model exhibits a separate queue for each part type. Workpieces of type i arrive with an exponential interarrival time or rate λ_i. The service time of a workpiece of a part type is given by an independent random variable of general distribution, denoted by B_i. The different part types are taken up for processing in the cyclic order $1, 2, \ldots, N-1, N, 1, 2$, etc. The processing of type i workpieces is preceded by a setup operation for part type i, given by a generally distributed duration R_i, which depends on part i, but is independent of other system parameters. When the machine has served waiting workpieces of type i (according to the given service policy), it changes over to the next part type $(i+1)$ modulo N by initiating a setup for this part type. Such a cyclic server or polling model has been solved exactly for the mean waiting time in queue of each part type, for the following cases.

(i) The service policy within a queue is the same for all the N part types, and can be exhaustive or gated. Within a queue, workpieces are processed in FCFS order

(ii) The special case where K-limited-exhaustive discipline is applied to the system for all the N part types, under the assumption that all the N parts have identical parameters, i.e., same arrival, setup and processing times (i.e., the *symmetric* case).

3.1a *Exhaustive service*: The system may be solved for the mean waiting times as follows: let b_i and $b_i^{(2)}$ denote the mean and second moment of the processing time for part i, B_i. Let $\rho_i = \lambda_i \times b_i$ be the utilisation of the machine by part type i, and let $\rho = \Sigma_{i=1}^{N} \rho_i$ be the total utilisation of the machine (the utilisation values exclude the time spent on setups). The mean of the total time spent on setup in each cycle is $R = \Sigma_{i=1}^{N} r_i$, and the variance of this time is $\Delta^2 = \Sigma_{i=1}^{N} \delta_i^2$.

The system is stable, i.e., the queue lengths for each part type do not build up to infinity, if $\rho < 1$.

The waiting time in queue of a type i workpiece is given by (Takagi 1988)

$$E[W_i] = E[I_i^2]/\{2E[I_i]\} + \lambda_i b_i^{(2)}/\{2(1-\rho_i)\},$$

where I_i denotes the intervisit time for the queue corresponding to part i. The intervisit time is defined as the duration starting the instant the machine leaves queue i and ending the instant the machine finishes setup for part i in the next cycle. The mean $E[I_i]$ and the variance $Var[I_i]$ of the intervisit time are given by

$$E[I_i] = (1-\rho_i)R/\{(1-\rho)\}$$

and

$$Var[I_i] = \delta_i^2 + \{(1-\rho_i)/\rho_i\} \sum_{j=1,\, j\neq i}^{N} r_{ij}$$

where $\{r_{ij}\}$, $1 \leqslant i, j \leqslant N$ are the set of covariances for the station times of the queues

for part types i and j. The station time for queue i is defined as the time interval between successive instants the machine starts setup for parts i and $i+1$. The values of the covariances r_{ij} are obtained by solving the set of the following $O(N^2)$ linear equations (Takagi 1988).

$$r_{ij} = \frac{\rho_i}{1-\rho_i}\left(\sum_{m=i+1}^{N} r_{jm} + \sum_{m=1}^{j-1} r_{jm} + \sum_{m=j}^{i-1} r_{mj}\right), \quad \text{for } j < i;$$

$$r_{ij} = \frac{\rho_i}{1-\rho_i}\left(\sum_{m=i+1}^{j-1} r_{jm} + \sum_{m=j}^{N} r_{mj} + \sum_{m=1}^{i-1} r_{mj}\right), \quad \text{for } j > i;$$

and

$$r_{ii} = \frac{\delta_i^2}{(1-\rho_i)^2} + \frac{\lambda_i b_i^{(2)} E[I_i]}{(1-\rho_i)^3} + \frac{\rho_i}{1-\rho_i}\left(\sum_{j=1, j\neq i}^{N} r_{ij}\right).$$

3.1b *Gated service*: We follow the same notation as in the exhaustive service case. The mean waiting times in queue are obtained by solution of a set of $O(N^2)$ linear equations. The mean waiting time for a workpiece of part type i is given by (Takagi 1988)

$$E[W_i] = (1+\rho_i)E[C_i^2]/\{2E[C]\},$$

where C_i is the random variable denoting the cycle time for part type i, defined as the time interval between successive instants of setup initiation for part type i. The expected value of the cycle time is independent of the part type and is given by

$$E[C] = E[C_i] = R/(1-\rho).$$

The condition for system stability is $\rho < 1$.

To obtain $E[C_i^2] = (E[C_i])^2 + Var[C_i]$, we solve for $Var[C_i]$ from:

$$Var[C_i] = \left(\frac{1}{\rho_i}\right)\sum_{j=1, j\neq i}^{N} r_{ij} + \sum_{j=1}^{N} r_{ji}.$$

The values $\{r_{ij}; 1 \leqslant i, j \leqslant N\}$ are again the set of covariances of station times. Here the station time for part type i is defined as the time interval between the successive instants the setups for part types i and $i+1$ are completed. The values of r_{ij} are obtained by solving the following set of linear equations (Takagi 1988)

$$r_{ij} = \rho_i\left(\sum_{m=i}^{N} r_{jm} + \sum_{m=1}^{j-1} r_{jm} + \sum_{m=j}^{i-1} r_{mj}\right), \quad \text{for } j < i;$$

$$r_{ij} = \rho_i\left(\sum_{m=i}^{j-1} r_{jm} + \sum_{m=j}^{N} r_{mj} + \sum_{m=1}^{i-1} r_{mj}\right), \quad \text{for } j > i;$$

and

$$r_{ii} = \delta_{i+1}^2 + \lambda_i \times b_i^{(2)} \times E[C] + \rho_i \times \sum_{j=1, j\neq i}^{N} r_{ij} + \rho_i^2 \times \sum_{j=1}^{N} r_{ji}.$$

3.1c *Limited service*: The limited service discipline has not been solved for the general asymmetric case (the solution is known for the symmetric case, Fuhrmann & Wang 1988). Several approximate approaches have been proposed in the literature; nearly all of them deal with the K-limited discipline with limit 1 for all the queues

(since the maximum number of workpieces of a particular part type processed in a cycle is limited to one, the K-limited-exhaustive and K-limited-gated variants become identical); the work reported in Ibe & Cheng (1989), Boxma & Meister (1987) and Srinivasan (1988) are representative of the approximate approaches to this special case of the problem. Fuhrmann & Wang (1988) present bounds for the computation of the mean waiting times under the more general situation where the limit on the number served is different for different part types; these bounds are extended to provide approximations to compute the mean waiting times. The survey articles by Takagi (1988, 1990) summarise the recent approximate analyses.

3.1d *Bernoulli scheduling*: The Bernoulli scheduling discipline constitutes a generalisation of the exhaustive and the K-limited-exhaustive service discipline with limit 1 for all part types; when p_i is 1 for all the N part types, it reduces to the exhaustive service, and when $p_i = 0$ for all parts it is the K-limited service discipline of limit 1. In the symmetric case, i.e., all the N part types have completely identical parameters, Tedijanto (1990) has shown that the mean waiting time of a workpiece can be given by

$$E[W] = \frac{1}{2(1-\rho-\lambda R(1-p))} \times \left(\lambda \sum_{j=1}^{N} b^{(2)} + \frac{(1-\rho)\Delta^2}{R} + R(1+\lambda b - 2\lambda b p) \right).$$

3.2 *Cyclic server models with limited buffers*

These models represent one important class, the general version of which is as yet unsolved. However, the single-buffer case, where each part type has a single buffer, has been successfully analysed; the recent results are given by Takine *et al* (1988), and Ibe & Cheng (1989). The solution of the symmetric version requires solving $O(2^{N-1})$ linear equations, and the general (asymmetric) version needs the solution of $O(2^{N-1})$ linear equations. Tran-Gia & Raith (1988) present an approximate analysis for more general systems where each part type may have a finite queue of (distinct) finite capacity, under non-exhaustive service assumptions. Seidmann *et al* (1985) have analysed a related model, in which a single server (a manufacturing cell) processes different types of part types, according to a probabilistic schedule and 1-limited service discipline; but they assume that at each service completion, a new workpiece is always available for each part type, i.e., no arrival process is explicitly considered; this assumption greatly simplifies the analysis.

4. An example

In this section, we consider the application of the various models to an example system – a single versatile machining centre processing three different part types. We restrict our attention to a few models due to space limitations. The mean values of the interarrival duration, setup time and processing time per workpiece for these part types are given in table 2.

Since the assumptions underlying the various models are different, the results produced by these will also differ; we shall concentrate on mean lead time (response time), or, equivalently, the mean inventory level (queue length) for each part type, and the mean utilisation.

Table 2. Parameters for the example.

	Mean time for		
Part type	Arrival	Processing	Setup
1	10·0	3·0	15·0
2	16·0	2·0	10·0
3	20·0	5·0	8·0

4.1 *M/G/1 queueing model*

In this model, the analyst assumes that all part types arrive with exponential interarrival times. All the parts wait in a common, infinite capacity queue from which they are taken up for processing in an FCFS fashion. Setup times are ignored. The mean lead time obtained is for a 'representative' part type. Let us consider the mean response time when the service time of each workpiece is k-stage Erlang with the mean value given above. A simple calculation gives the mean queue lengths when the service time has $k = 1, 2, 3$ stages (table 3).

A simple calculation gives the mean utilisations as $\rho_1 = 0{\cdot}3$, $\rho_2 = 0{\cdot}125$, and $\rho_3 = 0{\cdot}25$, and the total utilisation is $\rho = 0{\cdot}675$. An elementary application of Little's Law gives the mean lead times.

4.2 *GI/G/1 queue models*

We relax the exponential arrival assumption of the previous model and allow the arrival processes also to be GI. Other assumptions remain unchanged. The results are obtained for a typical part type which is representative of all the part types. Assume that the arrival and service processes of the three parts are all 1, 2 or 3 stage Erlang. Table 4 summarises the mean queue length values.

4.3 *Cyclic server queueing models*

Let us take a more detailed view of the example. Let parts 1 and 2 be members of a group technology part family, and part 3 be a member of another group technology part family. The workcentre produces them in the order 1, 2, 3 so that part family 1 is produced first, and then part family 2. Assume that arriving workpieces wait in a dedicated, infinite capacity queue. The processing of each part type is preceded by a setup for that part type. In such a case, the cyclic server multiqueue model of § 3 is

Table 3. Mean queue lengths, M/G/1 model.

Service time is		
1-Stage Erlang (exponential)	2-Stage Erlang	3-Stage Erlang
2·2442	1·85192	1·72115

Table 4. Mean queue lengths, GI/G/1 model.

	Service process		
Arrival process	1-Stage Erlang	2-Stage Erlang	3-Stage Erlang
1-Stage Erlang	2·2442	1·8519	1·7642
2-Stage Erlang	2·0543	1·6083	1·4897
3-Stage Erlang	1·9919	1·5263	1·4126

appropriate. Applying the exhaustive service policy would imply that all existing requirements of production for a particular part type are completed, and the machine changes over only when no further demand is present. Under the gated service policy, we process all outstanding demands for a particular part type that were present at the instant of setup completion. With limited service, we process each part until either a time limit expires or a maximum number has been processed, or there are no further waiting workpieces of that part type.

For simplicity, the setups are assumed to take an exponential duration. Let the machining times be k-Erlang for all parts; we consider $k = 1, 2, 3$. In table 5 we present the mean waiting times (this excludes the processing time) for each part type; the mean lead time is the sum of the mean waiting time and the mean processing time.

The machine utilisation values for the different part types are the same as before. For the 1-limited service discipline, the system is unstable, as an elementary check shows.

4.4 *Cyclic server model with single buffers*

Let there be N fixtures, one fixture each dedicated to each of the part types in the previous example; the infinite buffer assumption is no longer appropriate. We use the

Table 5. Mean waiting times, multiquene model.

Exhaustive service	Service time is		
Part type	1-Stage Erlang	2-Stage Erlang	3-Stage Erlang
1	47·66	45·96	45·39
2	50·98	57·68	56·91
3	51·60	49·81	49·21

Gated service	Service time is		
Part type	1-Stage Erlang	2-Stage Erlang	3-Stage Erlang
1	79·24	77·38	76·76
2	69·28	67·61	67·06
3	77·35	75·43	74·79

Table 6. Multiquene, single buffer model.

Performance measure	Part 1	Part 2·	Part 3
Throughput	0·0224	0·0205	0·0188
Maching utilisation	0·0672	0·0411	0·0939
Blocking probability (Mean queue length)	0·7662	0·6712	0·6244

cyclic server model where each part type has exactly one buffer (the fixture). When a fixture is occupied by a workpiece, further arrivals of workpieces of the same type are prohibited (loss system); when the fixture is released, a new arrival can take place. Assume, for simplicity, that all setup and processing times are exponential; also, ignore the fixturing time, or treat it as part of the processing time. Due to the single buffer per part assumption, exhaustive, gated and limited service disciplines become identical. We are interested in parameters like throughput of the parts (parts produced per unit time), the utilisation of the machine and the probability of an arrival finding the corresponding buffer full and hence being lost, for each part type. Under the assumption of independent exponential arrivals for each part type, applying the PASTA property (Wolff 1982), the probability that an arrival finds the corresponding buffers are full (and is hence lost), is the same as the steady state probability that the buffers are full. This result will hold even if the setup and processing activities have general i.i.d. durations; they need not be exponential. For our example, owing to the single buffer assumption, the probability that the buffer is occupied is the same as the mean number of workpieces of a particular part type in the system. We present these performance parameters for our example in table 6.

5. Conclusion

We considered a series of stochastic models for the analysis of a highly capable machining centre processing different types of parts. The issues addressed include using a single aggregate representative part type versus explicitly considering multiple part types, inclusion or non-inclusion of setup times, single common queue versus multiple queues (one for each part type), finite capacity queues versus infinite capacity queues, and exponential versus general arrival and service processes. Table 7 compares the degree of detail captured by each of the two major modelling approaches considered in this paper – (i) the GI/G/1 queue and variants (§ 2), (ii) the multiqueue cyclic server queueing model and variants (§ 3).

From a computational point of view, the GI/G/1 models are the simplest, since performance measures are easily determined by a straightforward computation using an explicit formula. The cyclic server systems (polling models with exponential arrivals) can be analysed by solving $O(N^2)$ linear equations. However, important variants like systems with mixed service, and systems with finite buffer capacities are solved only approximately.

The models considered in this paper can be used to analyse the performance of a single machine under varying demand rates, changing setup or processing times, or varying product mixes. In particular, the effect of introduction of new part types, or

Table 7. Comparison of the modelling approaches.

Issue	GI/G/1 queue	Multiqueue cyclic server
Part types	All part types aggregated to a single representative part type	N distinct parts
Setup time	Ignored	GI (explicit)
No. of queues	One common	One per part type
Queue capacity	Infinite (special cases like M/GI/1/N are solved)	Infinite (approximate solution for finite capacity case)
Arrivals	GI	Exponential (for exact solution)
Processing	GI	GI
Scheduling	FCFS	Cyclic service exactly solved; approximate solutions for others
Service policy	FCFS	Exhaustive, gated, limited

taking some part types out of production, on the lead times of parts already being manufactured can be studied. Besides, these models can form the building blocks of a more elaborate model of an AMS, consisting of several machines and material handling system equipment. Such a model of an AMS, will be typically solved by a decomposition approach (owing to the largeness of the system), by analysing individual equipment (e.g., deriving their flow equivalents) and combining these individual results into those of the entire system. The models proposed in this paper will enable explicit consideration of multiple part types and setup activities. Further, the introduction of the issues of machine failures and repair in the model, in the spirit of queueing systems with server vacations, will enhance the accuracy of the model. Development of solution methodologies for models of the AMS incorporating these features is part of our continuing work in this area.

References

Albin S L 1986 Delays for customers from different arrival streams to a queue. *Manage. Sci.* 32: 329–340

Ancker C Jr, Gafarian A V 1961 Queueing with multiple poisson inputs and exponential service times. *Oper. Res.* 9: 321–327

Basharin G 1965 A single server with a finite queue and items of several types. *Theory Probab. Its Appl. (USSR)* 10: 261–274

Baskett F, Chandy K M, Muntz R R, Palacios F G 1975 Open, closed and mixed networks of queues with different classes of customers. *J. Assoc. Comput. Mach.* 22: 248–260

Bitran G R, Tirupati D 1988 Multiproduct queueing networks with deterministic routing: Decomposition approach and the notion of interference. *Manage. Sci.* 34: 75–100

Bitran G R, Tirupati D 1989 Approximations for product departures from single-server station with batch processing in multi-product queues. *Manage. Sci.* 35: 851–878

Boxma O J, Meister B W 1987 Waiting-time approximations for cyclic-service systems with switchover times. *Performance Eval.* 7: 299–308

Buzacott J A, Shanthikumar J G 1985 On approximate queueing models of dynamic job shops. *Manage. Sci.* 31: 870–887

Dobson G, Karmarkar U S, Rummel J 1987 Batching to minimise flow times on one machine. *Manage. Sci.* 33: 784–799

Fuhrmann S W, Wang Y T 1988 Analysis of cyclic service systems with limited service: Bounds and approximations. *Performance Eval.* 9: 35–54

Gershwin S B 1989 Hierarchical flow control: A framework for scheduling and planning discrete events in manufacturing systems. *Proc. IEEE* 77: 195–209

Ibe O C, Cheng X 1989a Approximate analysis of asymmetric single-service token-passing systems. *IEEE Trans. Commun.* 37: 572–577

Ibe O C, Cheng X 1989b Performance analysis of asymmetric single-buffer polling systems. *Performance Eval.* 10: 1–14

Kelly F P 1979 *Reversibility and stochastic networks* (Chichester: John Wiley and Sons)

Kleinrock L, Levy H 1988 The analysis of random polling systems. *Oper. Res.* 36: 716–732

Kuehn P J 1979 Approximate analysis of general networks by decomposition. *IEEE Trans. Commun.* 27: 113–126

Kumar P R, Seidman T I 1990 Dynamic instabilities and stabilisation methods in distributed real-time scheduling of manufacturing systems. *IEEE Trans. Autom. Control.* 35: 289–298

Lavenberg S S 1975 The steady-state queueing time distribution for the M/G/1 finite capacity queue. *Manage. Sci.* 21: 501–506

Leachman R C, Gascon A 1988 A heuristic scheduling policy for multi-item single-machine production systems with time-varying, stochastic demands. *Manage. Sci.* 34: 377–390

Marie R A 1979 An approximate analytical method for general queueing networks. *IEEE Trans. Software Eng.* 5: 530–538

Perkins J R, Kumar P R 1989 Stable, distributed, real-time scheduling of flexible manufacturing/assembly/disassembly systems. *IEEE Trans. Autom. Control* 34: 139–148

Seidmann A, Schweitzer P J, Nof S Y 1985 Performance evaluation of a flexible manufacturing cell with random feedback flow. *Int. J. Product. Res.* 23: 1171–1184

Shanthikumar J G, Buzacott J A 1980 On the approximations to the single server queue. *Int. J. Product. Res.* 18: 761–773

Shanthikumar J G, Buzacott J A 1981 Open queueing network models of dynamic jobshops. *Int. J. Product. Res.* 19: 255–266

Shanthikumar J G, Gocmen M 1983 Heuristic analysis of closed queueing networks. *Int. J. Product. Res.* 21: 675–690

Srinivasan M M 1988 An approximation for mean waiting times in cyclic server systems with nonexhaustive service. *Performance Eval.* 9: 17–33

Takagi H 1988 Queueing analysis of polling models. *ACM Comput. Surv.* 20: 5–28

Takagi H 1990 Analysis of polling systems: An update. *Stochastic analysis of computer and communication systems* (Amsterdam: North Holland)

Takine T, Takahashi Y, Hasegawa T 1988 Exact analysis of asymmetric polling systems with single buffers. *IEEE Trans. Commun.* 36: 1119–1127

Tedijanto 1990 Exact results for the cyclic-service queue with a Bernoulli schedule. *Performance Eval.* 11: 107–115

Tran-Gia P, Raith T 1988 Performance analysis of finite capacity polling systems with nonexhaustive service. *Performance Eval.* 8: 1–16

Whitt W 1983 The queueing network analyser. *Bell Syst. Tech. J.* 62: 2779–2815

Whitt W 1988 A light-traffic approximation for single-class departure process from multi-class queues. *Manage Sci.* 34: 1333–1346

Wolff R W 1982 Poisson arrivals see time averages. *Oper. Res.* 30: 223–231

Queueing models with threshold phenomena: With special reference to machine interference problems

KARMESHU

School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi 110067, India

Abstract. Machine interference problems with state-dependent nonlinear repair rates depicting threshold behaviour are described. Stochastic analysis of the temporal evolution of the number of failed machines is possible by the 'diffusion approximation' method which can be systematically carried out to the desired degree of approximation. Thus one is able to study the system's widely differing stochastic behaviour near and away from the threshold point. Finally we point out the utility of this analysis to computer systems modelling.

Keywords. Machine interference problems; state-dependent nonlinear repair rate; threshold behaviour; diffusion approximation method; queueing models; stochastic behaviour.

1. Introduction

Models dealing with machine interference problems or equivalently finite-source queueing problems have gained significance in view of the increase in automation in production (Bhat 1984). Besides this, numerous applications in the information processing field have been analysed as finite-source queueing problems, one of the earliest applications being the machine repair model of time sharing systems (Lavenberg 1988). Since then several challenging applications have emerged in the computer communication field, which have led to the extension of queueing theory in general in new directions (Kleinrock 1988). Special mention may be made of incorporating nonlinearity, which in some cases manifests threshold/critical phenomena depicting sudden change (or discontinuous transition) in a system's output, for a small change in input value (Karmeshu & Jaiswal 1981; Kobayashi 1983; Nelson 1986, 1987; Kleinrock 1988). Existence of a critical state in the parameter space indicates 'change of states' as also regions of stability and instability. Nelson (1986, 1987) has studied this phenomenon within the framework of stochastic catastrophe theory with a view to modelling computer systems. Such a study can be used to suggest dynamic control policies in order to prevent sudden performance degradations.

Machine interference problems have usually been studied under the assumption that the repair rate is not affected by the number of failed machines. Karmeshu & Jaiswal (1981) have examined the situation when the increasing number of failed machines may either induce discouragement leading to lowering of repair rate or

urge the repairman to put in extra efforts to cope with the demand, thus increasing the repair rate. The outstanding feature of this model is the existence of the threshold effect. A detailed stochastic analysis of the system's behaviour in the vicinity and far from threshold point is presented in § 2 of this paper. In § 3 we discuss another model of machine interference with multiple steady states. The concluding section (§ 4) brings out the application and extension of these ideas to the information processing field. In the appendix we give a brief elementary introduction to both deterministic and stochastic catastrophe theories.

2. Threshold behaviour of nonlinear machine interference models

2.1 *The model and diffusion approximation*

Karmeshu & Jaiswal (1981) have proposed a model for the case when the repair rate μ_n depends nonlinearly on the number $n(t)$ of failed machines at time t. Specifically $\mu_n = N(\alpha N + \beta n)/(N + \gamma n)$, where N is the total number of machines, α is a positive constant and β and γ may be regarded as measures of encouragement and discouragement respectively. The well-known special case corresponding to linear (exponential) repair rate when there are enough repairmen can be obtained by setting $\beta = 0$ and $\gamma = 0$ (Tapiero 1988).

As usual the machines are assumed to be identical and the failure time is exponential with rate λ, so that the arrival rate when n machines have failed is $\lambda_n = \lambda(N - n)$. The process under study may be considered as a birth and death process for which transition rates are given by

$$\begin{aligned} &Pr\{n \to n+1 \ \text{in}(t, t+\delta t)\} = \lambda(N-n)\delta t + o(\delta t), \\ &Pr\{n \to n-1 \ \text{in}(t, t+\delta t)\} = [N(\alpha N + \beta n)/(\gamma n + N)]\delta t + o(\delta t), \\ &0 < n \leqslant N \text{ with } \lambda_0 = \lambda N,\ \mu_0 = 0. \end{aligned} \tag{1}$$

Although the time-dependent solution of the model cannot be found explicitly, useful results can be obtained for large N by the diffusion approximation technique. The time evolution of the system can, for large N, be approximately described by a stochastic differential equation (SDE) (see Arnold 1974, Gaver & Lehoczky 1977),

$$\begin{aligned} \mathrm{d}n(t) = &\{\lambda(N-n) - [N(\alpha N + \beta n)/(\gamma n + N)]\}\,\mathrm{d}t + [\lambda(N-n)]^{1/2}\,\mathrm{d}W_1(t) \\ &- [N(\alpha N + \beta n)/(\gamma n + N)]^{1/2}\,\mathrm{d}W_2(t), \end{aligned} \tag{2}$$

to be interpreted in the Ito sense. Here $[W_1(t), W_2(t)]$ is a two-dimensional Wiener process whose components are mutually independent.

To get an idea as to how (2) emerges on approximating from a birth–death process, we note that increment δn in the number of failed machines during a small time interval δt is a stochastic process with moments

$$\begin{aligned} &E[\delta n] = \{\lambda(N-n) - [N(\alpha N + \beta n)/(\gamma n + N)]\}\,\delta t, \\ &E[(\delta n)^2] = \{\lambda(N-n) + [N(\alpha N + \beta n)/(\gamma n + N)]\}\,\delta t. \end{aligned} \tag{2a}$$

While the expression for $E[\delta n]$ is that given by the corresponding deterministic model,

that for $E[(\delta n)^2]$ has the distinctive feature of being proportional to δt. Noting that the same features are possessed by the Wiener increment process $\mathrm{d}W(t)$ with

$$E[\mathrm{d}W(t)] = 0,\ E[\mathrm{d}W(t)]^2 = \mathrm{d}t,\ E[\mathrm{d}W(t)\,\mathrm{d}W(t')] = 0, \quad t \neq t',$$

the increment δn follows (2). Mathematical details can be found in Arnold (1974).

For large N, we get

$$n(t) = N\phi(t) + N^{\nu}x(t),\ 0 < \nu < 1, \tag{3}$$

where the first term on the right hand side represents the mean evolution of the system and the second term provides stochastic fluctuations about the mean (Dekker 1980). It may be pointed out that fluctuations are scaled as $O(N^{\frac{1}{2}})$ when the system is far from the threshold point and accordingly $\nu = \frac{1}{2}$ (van Kampen 1961). In a recent paper Jaiswal *et al* (1989) have carried out the stochastic analysis of the model in the vicinity of the threshold point, where the two parts of the stochastic process $n(t)$, viz. the deterministic term $N\phi(t)$, and the stochastic term $N^{\nu}x(t)$ turn out to be of the same order of magnitude. Consequently, the central limit theorem does not hold in the close neighbourhood of the threshold point and the system cannot be subdivided into uncorrelated subsets.

Substituting (3) into (2) and equating coefficients of leading terms on both sides, we obtain

$$\mathrm{d}\phi = [\lambda(1-\phi) - (\alpha + \beta\phi)/(1+\gamma\phi)]\,\mathrm{d}t, \tag{4}$$

and

$$\mathrm{d}x = [f_1(\phi)x + f_2(\phi)N^{\nu-1}x^2]\,\mathrm{d}t + [g(\phi)N^{\frac{1}{2}-\nu}]\,\mathrm{d}W_1 - [h(\phi)N^{\frac{1}{2}-\nu}]\,\mathrm{d}W_2, \tag{5a}$$

where

$$f_1(\phi) = (\alpha\gamma - \beta - \lambda - 2\lambda\gamma\phi - \lambda\gamma^2\phi^2)/(1+\gamma\phi)^2,$$
$$f_2(\phi) = (\beta - \alpha\gamma)/(1+\gamma\phi)^3,\ g(\phi) = [\lambda(1-\phi)]^{\frac{1}{2}}, \tag{5b}$$
$$h(\phi) = [(\alpha+\beta\phi)/(1+\gamma\phi)]^{\frac{1}{2}}.$$

Equation (4) gives the mean evolution of the system and (5) the fluctuations in the number of failed machines. The Fokker–Planck equation (FPE) corresponding to SDE (5a) is

$$\frac{\partial p(x,t)}{\partial t} = -\frac{\partial}{\partial x}[(f_1(\phi)x + f_2(\phi)N^{\nu-1}x^2)p(x,t)] + \frac{1}{2}(g^2(\phi) + h^2(\phi))N^{1-2\nu}\frac{\partial^2}{\partial x^2}p(x,t). \tag{6}$$

2.2 *Threshold effect*

The threshold feature emerges when $\alpha = \lambda$, $\beta \neq \alpha\gamma$. Following Karmeshu & Jaiswal (1981), the steady state ϕ_s is given by

$$\lambda(1-\phi_s) - (\lambda + \beta\phi_s)/(1+\gamma\phi_s) = 0. \tag{7}$$

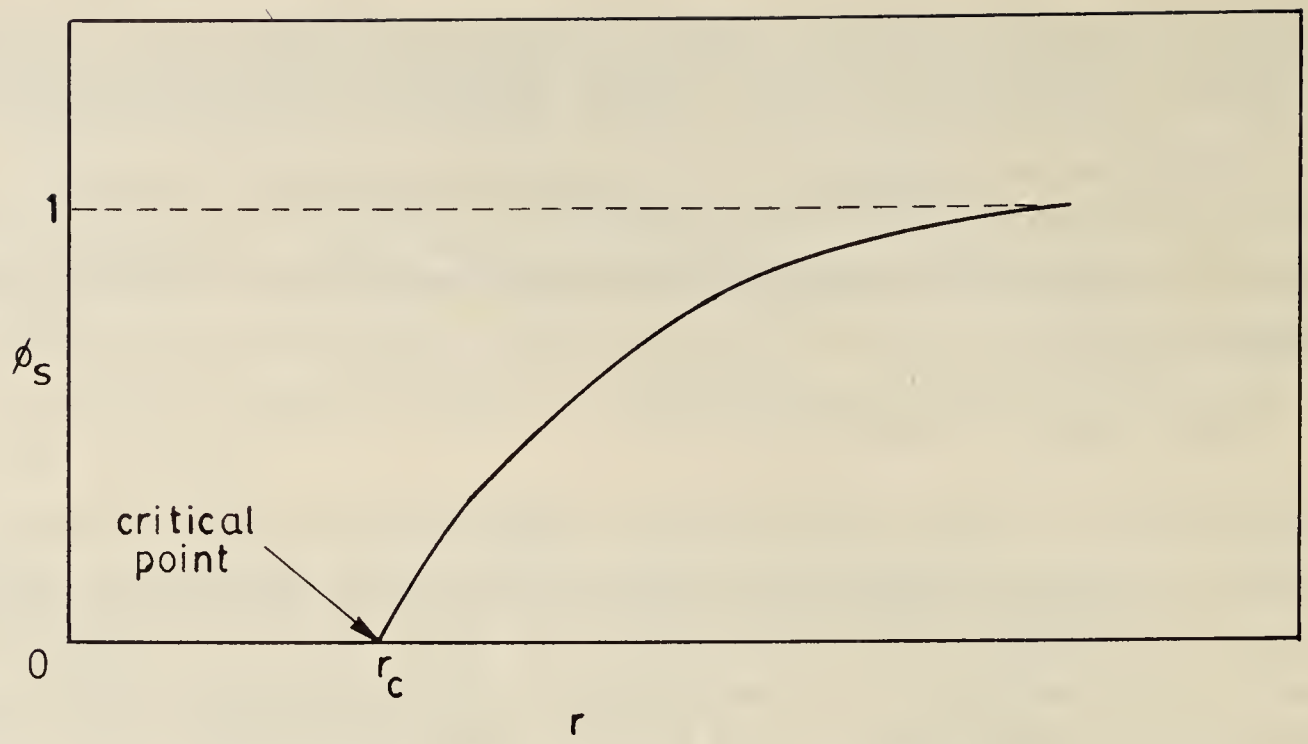


Figure 1. ϕ_s as a function of γ. The solid curve with singularity at $\gamma = \gamma_c$ depicts abrupt transition.

It follows that the steady state solution bifurcates at the critical point

$$\gamma = \gamma_c = 1 + \beta/\lambda, \tag{8a}$$

and the solution is

$$\phi_s = \begin{cases} 0, & \gamma \leqslant \gamma_c, \\ 1 - (\lambda + \beta)/\lambda\gamma, & \gamma > \gamma_c, \end{cases} \tag{8b}$$

Thus the critical/threshold point $\gamma = \gamma_c$ marks an abrupt transition from $\phi_s = 0$ to $\phi_s = 1 - (\lambda + \beta)/\lambda\gamma$ indicating that the average queue size $\bar{n}_s = N\phi_s$ will be zero until the control parameter γ exceeds the value γ_c (see figure 1). The other solutions $\phi_s = 0$ for $\gamma > \gamma_c$ and $\phi_s = 1 - (\lambda + \beta)/\lambda\gamma$ for $\gamma > \gamma_c$ are unstable. It may be mentioned in passing that this model describes *a situation which is an example of fold catastrophe* (Nelson 1987). A fold has a potential function that is cubic and thus can have no stable equilibrium points or can have exactly one. The threshold point $\gamma = \gamma_c$ corresponds to the first spot where the potential function ceases to have an inflection point and instead has one local minima and maxima. Such thresholds are characteristics of fold catastrophes.

2.3 *Stochastic analysis*

We now summarize the two situations discussed by Jaiswal *et al* (1989).

(i) *Far from threshold point*: Far away from the threshold point, the conventional scaling $\nu = \frac{1}{2}$ applies and the FPE (6) has the asymptotic limit

$$\frac{\partial p}{\partial t} = -f_1(\phi)\frac{\partial}{\partial x}(xp) + \frac{1}{2}[g^2(\phi) + h^2(\phi)]\frac{\partial^2 p}{\partial x^2}, \tag{9}$$

which represents the well-known nonstationary Ornstein–Uhlenbeck process with Gaussian density $p(x, t)$.

(ii) *Threshold region*: The behaviour in the vicinity of the threshold point $\gamma = \gamma_c$ is very different from that mentioned above. It may be noted from (6) that $f_1(\phi)$ vanishes at $\gamma = \gamma_c$. The drift and diffusion coefficients in FPE turn out to be $O(N^{\nu-1})$ and $O(N^{1-2\nu})$ respectively and they are comparable if $\nu - 1 = 1 - 2\nu$ or $\nu = 2/3$, when (6) reduces to

$$\frac{\partial p}{\partial t} = N^{-1/3}\left[-\frac{\partial}{\partial x}\{f_1(\phi)xN^{1/3} + f_2(\phi)x^2\}p + \frac{1}{2}\{g^2(\phi) + h^2(\phi)\}\frac{\partial^2 p}{\partial x^2}\right] \tag{10}$$

It is instructive to note that if $\gamma - \gamma_c = O(N^{-1/3})$ and $\phi_s = O(N^{-1/3})$ then all terms in (10) are of the same order of magnitude and may, therefore, be taken to define the threshold region around the threshold point $\gamma = \gamma_c$. The factor $N^{-1/3}$ on the right hand side of (10) signifies the well-known phenomenon of critical slowing down.

Observing that 0 and ∞ are natural boundaries, the steady state probability density is found to be

$$p_s(x) = \text{const. } \exp[-\tfrac{1}{2}(\gamma - \gamma_c)N^{1/3}x^2 - \tfrac{1}{3}(1 + \gamma - \gamma_c)x^3] \tag{11}$$

which at the critical point reduces to

$$p_c(x) = \text{const. } \exp[-\tfrac{1}{3}x^3], \quad 0 \leqslant x < \infty. \tag{12}$$

Introducing the variable $\rho = n/N = \phi + N^{-1/3}x$ and setting $N(\gamma - \gamma_c) = \eta$, (11) gives, for $|\eta| \ll 1$, in the zero order approximation:

$$\frac{\partial \rho_s}{\partial \eta} = \tfrac{1}{2}(\overline{\rho_s^3} - \overline{\rho_s}\,\overline{\rho_s^2}). \tag{13}$$

This equation shows the manner in which the rate of growth of $\overline{\rho_s}$ with respect to η is related to the first three moments of the distribution. This also elucidates the role played by fluctuations in bringing about the phenomenon. An interesting feature of this model is the existence of a critical state in the parameter space and the build up of fluctuations in its neighbourhood.

3. Machine interference model with multiple steady states

3.1 *The model and diffusion approximation*

In the machine interference problem another salient feature of the existence of multiple steady states may emerge when the repair rate increases in the initial stage but starts to decline as the number of failed machines increases further. It would be worth mentioning here that this type of service/repair rates frequently arise in problems related to computer performance models such as multiprogramming in computer systems, buffer control in communication networks (Nelson 1987). In the model of multiprogramming in computer systems discussed by Nelson (1987), the number of customers in the wait queue and CPU/IO system is a Markov process with arrival rate $\lambda(N - n)$ and service rate $\alpha(n)$ (or $\mu(n)$). For this the CPU/IO subsystem is to be replaced by a 'flow equivalent' single server queue with state-dependent service rate $\alpha(n)$ (or $\mu(n)$), $0 \leqslant n \leqslant N$. Nelson (1987) has shown that this model exhibits stochastic cusp catastrophe.

In a recent paper, Jaiswal *et al* (1990) have considered the machine interference problem when the repair rate increases with n upto a certain stage but starts declining as n increases further. Accordingly, the repair rate μ_n can be taken as

$$\mu_n = N[\alpha + \beta(n/N) + \alpha\delta(n/N)^2]/[1 + \delta(n/N)^2],$$

with parameters α, β, δ being positive. The parameters β and δ may be regarded as measures of encouragement and discouragement respectively.

As discussed in § 2, the number of failed machines $n(t)$ is a birth and death process with infinitesimal probabilities:

$$Pr\{n \to n+1 \text{ in } (t, t+\delta t)\} = \lambda_n \delta t + o(\delta t) = \lambda(N-n)\delta t + o(\delta t)$$

$$Pr\{n \to n-1 \text{ in } (t, t+\delta t)\} = \mu_n \delta t + o(\delta t)$$

$$= \{N[\alpha + \beta(n/N) + \alpha\delta(n/N)^2] \div$$

$$\div [1 + \delta(n/N)^2]\}\, \delta t + o(\delta t). \qquad (14)$$

It is interesting to note that the form of μ_n does have the correct shape to reproduce the characteristics of a cusp catastrophe.

The functions λ_n and μ_n are graphed in figure 2. It is easy to see that the μ_n-curve intersects the λ_n-straight line at three points in the following two situations:

(i) $\lambda_{\min} < \lambda < \lambda_{\max}$ when α, β and δ have a given set of values.
(ii) $\alpha_{\min} < \alpha < \alpha_{\max}$ when λ, β and δ have a given set of values.

Whenever λ is outside the interval $(\lambda_{\min}, \lambda_{\max})$ or α is outside the interval $(\alpha_{\min}, \alpha_{\max})$,

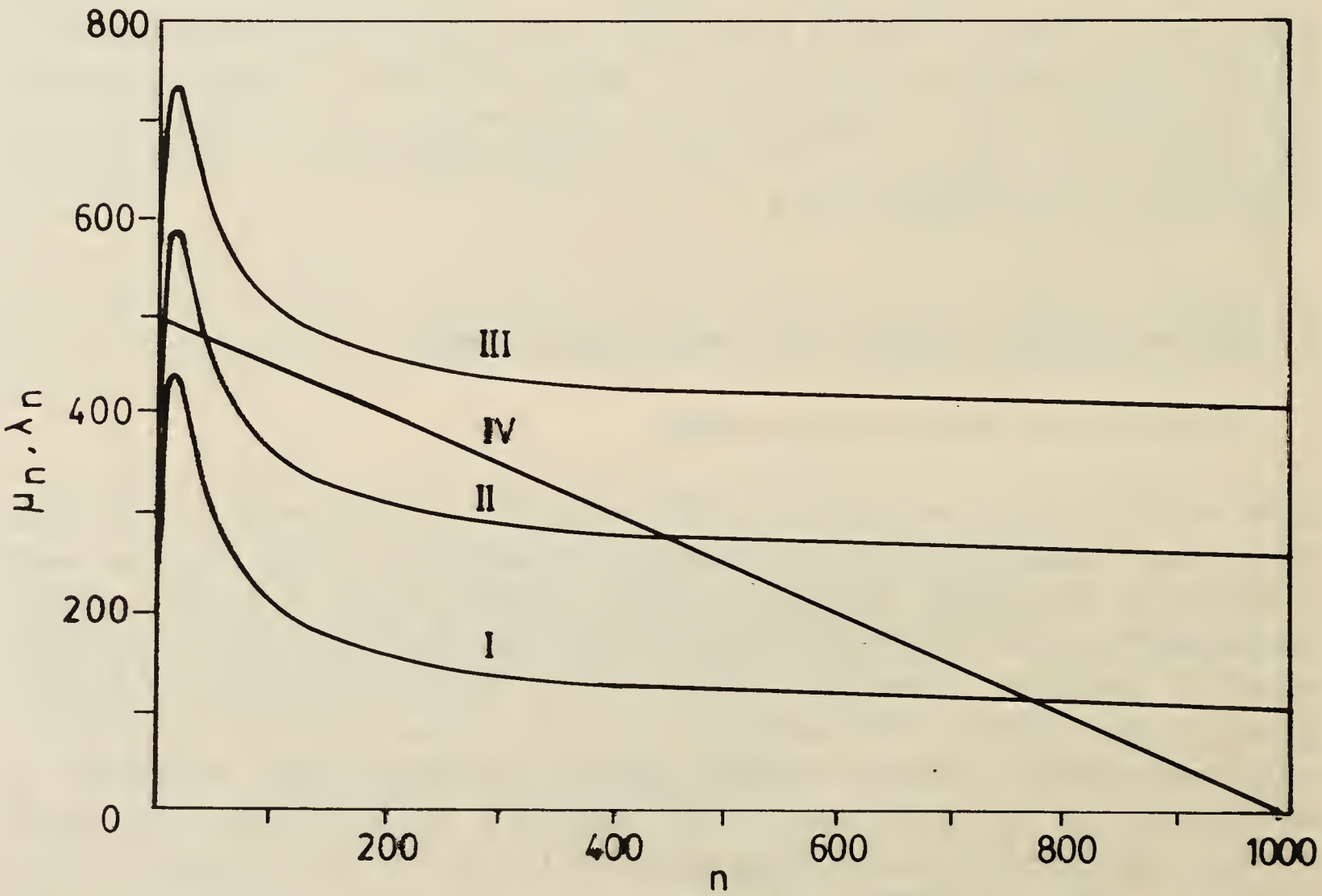


Figure 2. $N = 1000$, $\beta = 40$, $\delta = 3500$. Curves I, II, III depict μ_n for $\alpha = 0{\cdot}1$, $0{\cdot}25$ and $0{\cdot}4$ respectively. Straight line IV depicts μ_n for $\lambda = 0{\cdot}5$, its intersections with the curves represent steady states.

λ_n and μ_n curves intersect only at one point. Thus in the steady state three values of $\phi_s = n_s/N$ are possible. In principle either α or λ can be taken as a control variable while examining the multiple steady states of the state variable ϕ_s.

The temporal evolution of the system for large N can be investigated through SDE.

$$\mathrm{d}n(t) = \left\{\lambda N\left(1-\frac{n}{N}\right) - N\left[\alpha+\beta\frac{n}{N}+\alpha\delta\left(\frac{n}{N}\right)^2\right]\Big/\left[1+\delta\left(\frac{n}{N}\right)^2\right]\right\}\mathrm{d}t + \\ + \left[\lambda N\left(1-\frac{n}{N}\right)\right]^{\frac{1}{2}}\mathrm{d}W_1 - \left\{N\left[\alpha+\beta\frac{n}{N}+\alpha\delta\left(\frac{n}{N}\right)^2\right]\div \right. \\ \left. \div\left[1+\delta\left(\frac{n}{N}\right)^2\right]\right\}^{\frac{1}{2}}\mathrm{d}W_2, \qquad (15)$$

where $(W_1(t),\ W_2(t))$ is a two-dimensional Wiener process with independent components. Applying the diffusion approximation technique (as discussed in §2) when the system is far from the threshold point, we set for large N,

$$n(t) = N\phi(t) + N^{\frac{1}{2}}x(t). \qquad (16)$$

It is assumed that $E[n(t)]$ for large N is $O(N)$ and the fluctuations around this mean path are $O(N^{\frac{1}{2}})$. Substituting (16) into (15) and equating coefficients of leading terms on both sides, we obtain the differential equations for the mean evolution and stochastic fluctuations in the system respectively; these are

$$\mathrm{d}\phi/\mathrm{d}t = \{\lambda(1-\phi) - [(\alpha+\beta\phi+\alpha\delta\phi^2)/(1+\delta\phi^2)]\} \qquad (17)$$

$$\mathrm{d}x = -[\lambda+\beta(1-\delta\phi^2)/(1+\delta\phi^2)]x\,\mathrm{d}t + [\lambda(1-\phi)]^{\frac{1}{2}}\mathrm{d}W_1 - \\ -[(\alpha+\beta\phi+\alpha\delta\phi^2)/(1+\delta\phi^2)]^{\frac{1}{2}}\mathrm{d}W_2. \qquad (18)$$

3.2 *Multiple steady-state solutions*

The time-dependent solution of (17) can be expressed in the form

$$[(\phi-\phi_1)/(\phi_0-\phi_1)]^{a_1}[(\phi-\phi_2)/(\phi_0-\phi_2)]^{a_2}[(\phi-\phi_3)/(\phi_0-\phi_3)]^{a_3} \\ = \exp[-\lambda\delta(\phi_1-\phi_2)(\phi_2-\phi_3)(\phi_3-\phi_1)t], \qquad (19)$$

where $\phi_0 = n_0/N$ is the initial value of ϕ and ϕ_1, ϕ_2, ϕ_3 are steady state solutions of (17); also

$$a_1 = (\phi_3-\phi_2)(1+\delta\phi_1^2),\ a_2 = (\phi_1-\phi_3)(1+\delta\phi_2^2), \\ a_3 = (\phi_2-\phi_1)(1+\delta\phi_3^2). \qquad (20)$$

For physical relevance the steady state solutions ϕ_1, ϕ_2, ϕ_3 should be positive. Without loss of generality we assume $\phi_1 \leqslant \phi_2 \leqslant \phi_3$. It follows immediately from (19) that for large values of t, $\phi \to \phi_1$ or ϕ_2 or ϕ_3 according to whether ϕ_0 is less than, equal to, or greater than ϕ_2.

In the steady state, (17) reduces to

$$\phi_s^3 - \left(1-\frac{\alpha}{\lambda}\right)\phi_s^2 + \left(\frac{\beta}{\lambda}+1\right)\frac{1}{\delta}\phi_s - \left(1-\frac{\alpha}{\lambda}\right)\frac{1}{\delta} = 0. \qquad (21)$$

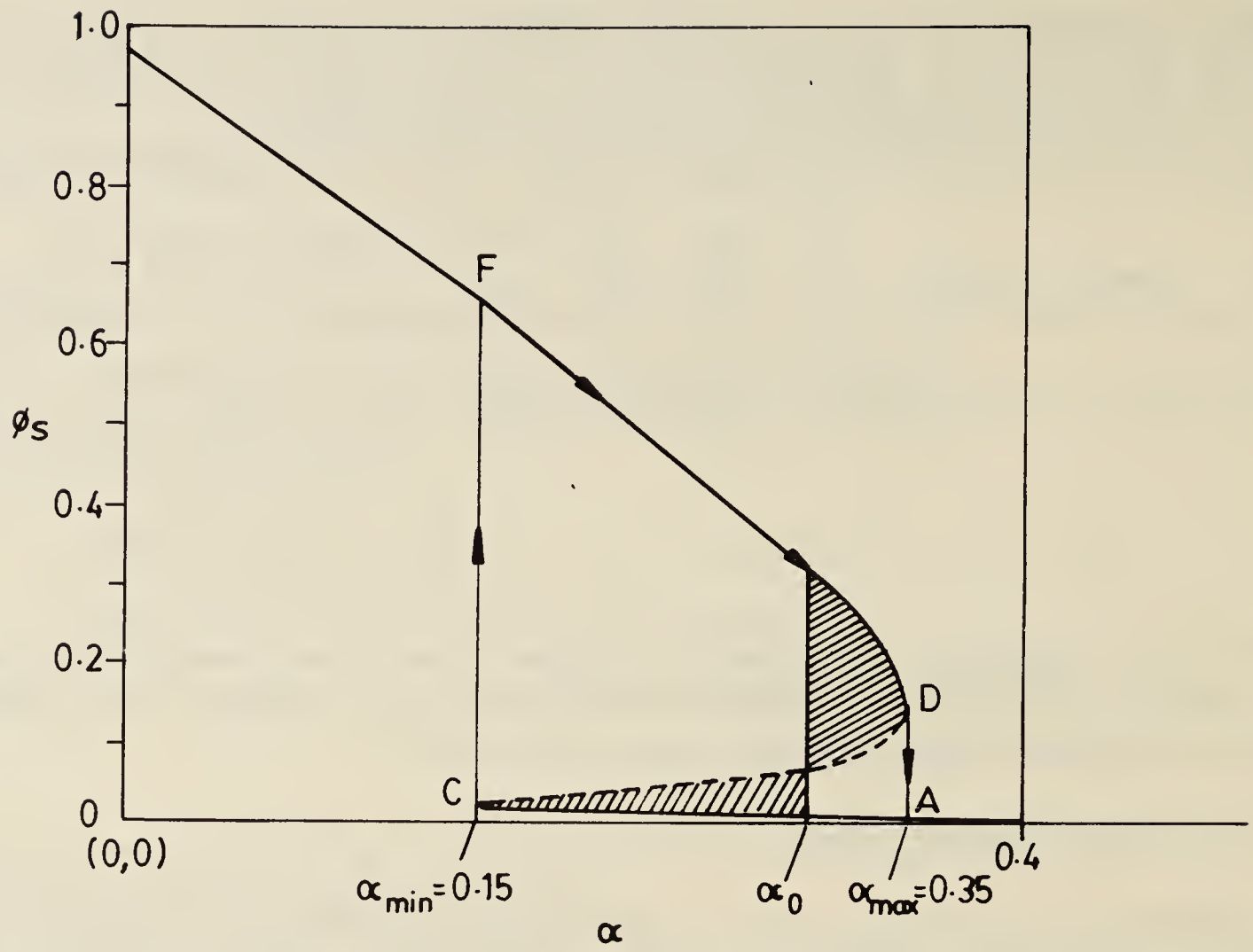


Figure 3. $\lambda = 0{\cdot}5$, $\beta = 40$, $\delta = 3500$, $\alpha_{max} = 0{\cdot}15$, $\alpha = 0{\cdot}35$.

Evidently when $\alpha = \lambda$, $\phi_s = 0$ is the only steady state. When α decreases from higher values to α_{max}, then for certain values of β and δ, the only value of ϕ_s gradually increases and when $\alpha = \alpha_{max}$, ϕ_s takes three values of which two are coincident. As α decreases further ϕ_s has three distinct values and when $\alpha = \alpha_{min}$ two of the values coincide. Finally, when $\alpha < \alpha_{min}$ there is only one value of ϕ_s. Thus for $\alpha_{min} < \alpha < \alpha_{max}$, ϕ_s carves out an inverted *S*-curve. These results are shown in figure 3. Details of calculating α_{min} and α_{max}, as also ϕ_s, are given in the paper by Jaiswal *et al* (1990).

Carrying out linear stability analysis it can be shown that in figure 3 the lower branch AC and the upper branch DF correspond to stable states and the middle branch CD to unstable states which cannot be realized. Further, from figure 3 we observe that when α starts decreasing from a value greater than α_{max}, the steady state value lies on the lower branch AC until $\alpha = \alpha_{min}$ and as α decreases further, the steady state ϕ_s jumps (abruptly) to the upper branch. On the other hand, when α starts increasing from a value less than α_{min}, the state ϕ_s moves on the upper branch until α attains the value α_{max}. As α increases further, ϕ_s jumps down to the lower branch. This salient feature is known as the hysteresis effect. In passing, it may be pointed out that the situation explained by this model is an example of cusp catastrophe.

3.3 *Stochastic analysis*

Our model admits of one steady state ϕ_s when $\beta < 8\lambda$ and three when $\beta > 8\lambda$. Hence from the discussion of § 3.1 the probability density function (PDF) of the number of failed machines in the steady state will be unimodal, having a sharp peak at $N\phi_s$. When $\beta > \delta\lambda$, there are three steady state values of ϕ_s of which the two stable values are modes of a bimodal PDF and in this case $\alpha_{min} < \alpha < \alpha_{max}$. We now give briefly the

derivation of the actual probability density functions from the paper by Jaiswal *et al* (1990).

Case I. $\beta < 8\lambda$: Equation (21) has only one real root and the FPE corresponding to (18) is

$$\frac{\partial p}{\partial t} = -\frac{\mathrm{d}}{\mathrm{d}\phi}A(\phi)\frac{\partial}{\partial x}(xp) + \frac{1}{2}B(\phi)\frac{\partial^2 p}{\partial x^2}, \tag{22}$$

where

$$A(\phi) = [\lambda - \alpha - (\lambda + \beta)\phi + (\lambda - \alpha)\delta\phi^2 - \lambda\delta\phi^3]/(1 + \delta\phi^2),$$

and

$$B(\phi) = [\lambda + \alpha + (\beta - \lambda)\phi + (\lambda + \alpha)\delta\phi^2 - \lambda\delta\phi^3]/(1 + \delta\phi^2). \tag{23}$$

Equation (22) governs the statistical fluctuations in the system and corresponds to a non-stationary Ornstein–Uhlenbeck process. The solution of the Fȯkker–Planck equation (22) is the gaussian distribution

$$p(x, t) = [2\pi\sigma_x^2]^{-\frac{1}{2}} \exp[-x^2/2\sigma_x^2]$$

with

$$\sigma_x^2(t) = A^2(\phi)\int_{\phi_0}^{\phi(t)} [B(u)/A^3(u)]\,\mathrm{d}u, \tag{25}$$

the initial distribution being the Dirac delta function

$$p(x, 0) = \delta(x). \tag{26}$$

The original stochastic variable $n(t)$ is, therefore, governed by the PDF

$$p(n, t) = \{1/[(2\pi N)^{\frac{1}{2}}\sigma_x(t)]\} \exp\{-[n - N\phi(t)]^2/[2N\sigma_x^2(t)]\}. \tag{27}$$

The PDF corresponding to FPE (22) in the steady state is

$$p_s(x) = \text{const. } \exp[(A'(\phi_s)/B(\phi_s))x^2]. \tag{28}$$

Case II. $\beta > 8\lambda$: Now $\alpha_{\min} < \alpha < \alpha_{\max}$, the steady state probability density functions at the lower branch $\phi_s = \phi_1$ and the upper branch $\phi_s = \phi_3$ are

$$p_s^{(i)}(x) = \text{const. } \exp\left[-\frac{[3\lambda\delta\phi_i^2 - 2\delta(\lambda - \alpha)\phi_i + \lambda + \beta]x^2}{[(\lambda - \alpha)\delta - \lambda - \alpha]\phi_i^2 - 2\beta\phi_i - 2\alpha}\right], \quad i = 1, 3, \tag{29}$$

with

$$\begin{aligned}
\phi_1 &= \tfrac{1}{3}b + 2\rho^{1/3}\cos[\tfrac{1}{3}(2\pi + \theta)],\\
\phi_3 &= \tfrac{1}{3}b + 2\rho^{1/3}\cos[\tfrac{1}{3}(4\pi + \theta)],\\
b &= 1 - (\alpha/\lambda),\ \rho^2 = -H^3,\ \theta = \tan^{-1}(\Delta/G),\ H = (1/3)r - (1/9)b^2,\\
G &= (1/3)b[r - (1/\delta)] - (2/27)b^3,\ r = [1 + (\beta/\lambda)](1/\delta),\\
\Delta^2 &= -(G^2 + 4H^3).
\end{aligned} \tag{30}$$

The density functions (29) are each locally gaussian and when combined determine

a bimodal distribution for $\alpha_{min} < \alpha < \alpha_{max}$. In the context of backlog in slotted ALOHA, Kleinrock (1988, p. 32) has obtained similar unimodal and bimodal distributions.

4. Conclusions

Nelson (1986, 1987) in a series of papers has argued that the utility of employing stochastic catastrophe theory to computer system modelling stems from the fact that the theory enables one to study a variety of systems which have common behavioural characteristics. He has analysed models of slotted ALOHA in broadcast networks, multiprogramming in computer systems, and buffer flow control in communication networks. Since these models have similar characteristics with regard to bimodality, sudden jumps, hysteresis etc. they can be described as examples of stochastic cusp catastrophe. This brings out the importance of stochastic catastrophe theory being used in synthetically interpreting computer system modelling.

It is interesting to note that the problems of multiprogramming, finite population slotted ALOHA etc. can be viewed as machine interference problems with state-dependent repair rate. The form of $\mu(n)$ considered in the previous section does have the correct shape to reproduce the characteristics of a cusp catastrophe. An important question is whether there is probabilistic interpretation of this form. To derive a form of $\mu(n)$ based on probabilistic considerations is a matter for future enquiry.

In passing we remark that the critical region around the threshold point in the machine interference problem with multiple steady states is under investigation. This study should yield a bimodal PDF unlike the one we have obtained by combining two locally gaussian probability density functions.

Appendix A

This appendix is primarily based on the paper by Nelson (1987) and gives a brief elementary introduction of deterministic catastrophe theory and stochastic catastrophe theory.

Deterministic catastrophe theory

It is the shape of the potential function which characterizes the behaviour of the system and we consider a system governed by the potential function $V(x, \lambda)$ with control parameters $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$. The critical points given by $\partial V/\partial x = 0$ correspond to three types of equilibrium states – stable, unstable and degenerate, according to whether $\partial^2 V/\partial x^2 >$, $<$ or $= 0$, the last state marking out a boundary between the former two and is called a bifurcation state. For a given value of λ, the drift of the system is given by $-\partial V(x, \lambda)/\partial x$.

Usually equilibrium states change gradually as the parameter λ changes but there may be certain values λ_c of λ for which a small change gives a sudden and possibly a violent jump from one equilibrium position to another. These sudden transitions are called catastrophes and the catastrophe set K comprises some coalescent points at which it is possible for the maximum and minimum of $V(x, \lambda)$ to coalesce and disappear (Chillingworth 1976). Catastrophe theory provides a method of finding the number of essential parameters of a potential function from its Taylor expansion

(Poston & Stewart 1987). Thom's theorem classifies the shapes of scalar potential functions locally about their critical points. The types of potential functions usually met with are cuspoids and a partial list is given below:

(1) the Fold: $V(x, \lambda_1) \equiv x^3 + \lambda_1 x$;
(2) the Cusp: $V(x, \lambda_1, \lambda_2) \equiv x^4 + \lambda_1 x^2 + \lambda_2 x$;
(3) the Swallow tail: $V(x, \lambda_1, \lambda_2, \lambda_3) \equiv x^5 + \lambda_1 x^3 + \lambda_2 x^2 + \lambda_3 x$
(4) the Butterfly: $V(x, \lambda_1, \lambda_2, \lambda_3, \lambda_4) \equiv x^6 + \lambda_1 x^4 + \lambda_2 x^3 + \lambda_3 x^2 + \lambda_4 x$.

Stochastic catastrophe theory

In the context of deterministic catastrophe theory it is the shape of the potential function which characterises the behaviour of the system. Several researchers (e.g. Haken 1983, Nelson 1986, 1987) have extended the framework of catastrophe theory to include stochastic features.

Stationary probability density function (PDF) of a stochastic process plays the same role that a potential function does in a deterministic process. One expects a high probability of finding a deterministic system about a local minimum of its potential function (corresponding to a stable equilibrium) and for the corresponding stochastic process, the same is true if its stationary PDF has a local maximum. These facts are borne out in the diagrams (figure A1) reproduced from Haken (1983).

In stochastic catastrophe theory one can define c-potential analogously to that of potential function of a deterministic process. We consider a stochastic process $X(t)$

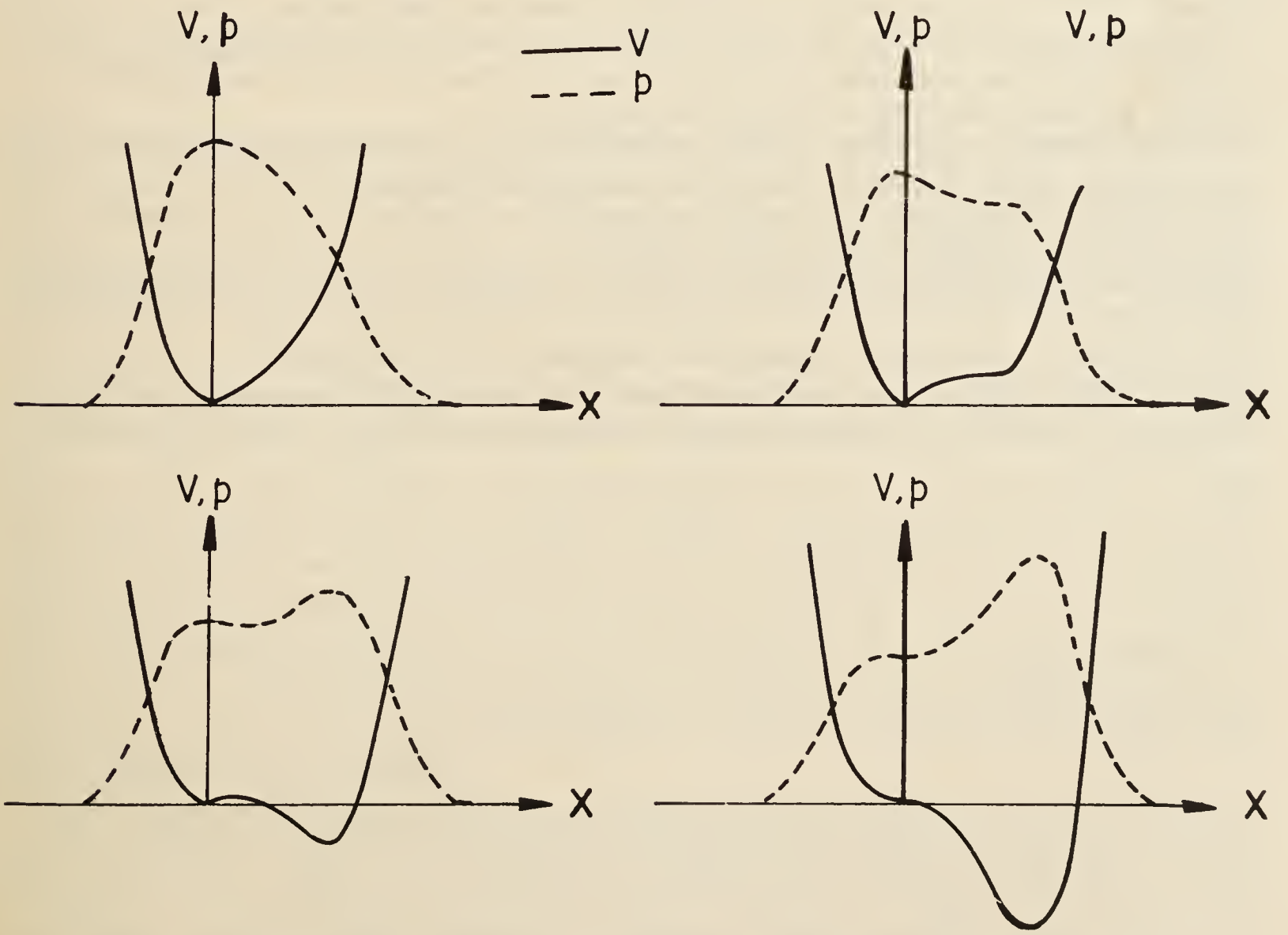


Figure A1. Graphs of potential curves V (solid) and probability density functions p against x for four different values of control parameter.

with infinitesimal drift $\mu(x)$ and infinitesimal variance $\sigma^2(x)$. The c-potential function is defined as

$$V(x) = -\int^x [2\mu(y)/\sigma^2(y)]\,dy.$$

To quote Nelson (1987), 'this definition is meaningful only if behaviour, analogous to a deterministic catastrophe, is found in diffusion processes when their c-potential function is equivalent to one of Thom's canonical formulas'.

References

Arnold L 1974 *Stochastic differential equations: Theory and applications* (New York: Wiley)

Bhat U N 1984 *Elements of applied stochastic processes* 2nd edn (New York: Wiley)

Chillingworth D R J 1976 Structural stability of mathematical models. In *Mathematical modelling* (eds) J G Andrews, R R Mclone (London: Butterworths)

Dekker H 1980 On the critical point of a Malthus-Verhulst process. *J. Chem. Phys.* 72: 189–192

Gaver D P, Lehoczky J P 1987 A diffusion approximation solution for a repairmen problem with two types of failure. *Manage. Sci.* 24: 71–81

Haken H 1983 *Synergetics: An introduction* 3rd edn (New York: Springer-Verlag)

Jaiswal M P, Sharma C L, Karmeshu 1989 Stochastic analysis of a non-linear machine interference model. *Math. Comput. Simulation* (in press)

Jaiswal M P, Sharma C L, Karmeshu 1990 Stochastic analysis of a finite source queueing model with hysteresis effect. *Math. Comput. Simulation* (submitted)

Karmeshu, Jaiswal N K 1981 A machine interference model with threshold effect. *J. Appl. Probab.* 18: 491–498

Kleinrock L 1988 Performance evaluation of distributed computer-communication systems: In *Queueing theory and its applications* (eds) O J Boxma, R Syski (Amsterdam: North Holland)

Kobayashi H 1983 Stochastic modelling: Queueing models. In *Probability theory and computer science* (eds) G Louchard and G Latouche (New York: Academic Press)

Lavenberg S S 1988 A perspective on queueing models of computer performance. In *Queueing theory and its applications* (eds) O J Boxma, R Syski (Amsterdam: North Holland)

Nelson R 1986 How to control those unruly-tape measures or computer performance modelling using stochastic catastrophe theory. *Math. Intell.* 8: 50–56

Nelson R 1987 Stochastic catastrophe theory in computer performance modelling. *J. Assoc. Comput. Mach.* 34: 661–685

Poston T, Stewart I 1978 *Catastrophe theory and its applications* (London: Pitman)

Tapiero C S 1988 *Applied stochastic models and control in management* (Amsterdam: North-Holland)

van Kampen N G 1961 A power series expansion of the master equation. *Can. J. Phys.* 39: 551–567

Integrated analytical models for flexible manufacturing systems

Y NARAHARI, N VISWANADHAM, C R MEENAKSHISUNDARAM and P HANUMANTHA RAO

Department of Computer Science & Automation, Indian Institute of Science, Bangalore 560012, India

Abstract. Product form queueing networks (PFQN) and generalized stochastic Petri nets (GSPN) have emerged as the principal performance modelling tools for flexible manufacturing systems (FMS). In this paper, we present integrated PFQN–GSPN models, which combine the computational efficiency of PFQN and representational power of GSPN by employing the principle of flow-equivalence. We show that FMS that include non-product form characteristics such as dynamic routing and synchronization can be evaluated efficiently and accurately using the integrated models.

Keywords. Integrated analytical models; flexible manufacturing systems; product form queueing networks; generalised stochastic Petri nets.

1. Introduction

Flexible manufacturing systems (FMS) (Ranky 1983) represent an important recent development in manufacturing automation, on account of improved productivity, quality and resilience to demand fluctuations. An FMS is essentially a discrete event dynamic system (Ho 1987) and therefore Markov chains constitute the basic stochastic model of FMS. Queueing networks (QN) (Buzacott & Yao 1986) and generalized stochastic Petri nets (GSPN) (Viswanadham & Narahari 1988) have emerged as the principal analytical modelling tools for FMS. Product form queueing network (PFQN) models of FMS have gained prominence because of their computational efficiency, however, they cannot capture FMS characteristics such as priority decision rules, blocking of resources, dynamic routing, multiple resource holding, and synchronization. To model these (non-product form) features, researchers have often employed approximations (Chandy & Sauer 1978). GSPN provide an exact way of modelling non-product form features and as natural models of concurrency and synchronization, GSPN have emerged as a strong alternative to PFQN in many modelling situations. However, as FMS grow in size and complexity, the GSPN models lead to large state space, making them intractable.

The concept of flow-equivalence in queueing networks (Chandy *et al* 1975) can be used in building realistic and efficiently solvable performance models that include two parts: one a queueing network part and the other a Petri net (PN) part. In particular, we shall consider models that include a PFQN part and a GSPN part. Such

models were first considered by Balbo *et al* (1986) to evaluate computer systems with non-product form features such as priorities, blocking, forking and joining, dynamic routing, and multiple resource holding. Meenakshisundaram (1990) and Ramesh (1990) have looked at the application of this modelling technique in the context of parallel and distributed computing systems. In this paper, we bring out the use of such models, which we shall refer to as integrated PFQN-GSPN models (IQP models for short) in the FMS context.

In the next section, we introduce the notation of GSPN and explain the basic mechanism of IQP modelling. In § 3, we present two illustrative examples. All the examples are concerned with the closed central server models of FMS. In example 1, we present an IQP model of a central server FMS where parts are routed to machines dynamically using the SQR (shortest queue routing) policy. In example 2, we consider a central server FMS in which fresh raw parts are released into the FMS only when an explicit demand is pending for a finished part and develop an IQP model. In § 4, we discuss the accuracy of IQP modelling and show with numerical results that the performance estimates are remarkably accurate.

2. Integrated PFQN–GSPN modelling

Here, we first introduce GSPN and then explain integrated PFQN-GSPN models.

2.1 *Generalized stochastic Petri nets*

In the following definitions (Murata 1989), N denotes the set of non-negative integers.

DEFINITION 1

A Petri net G is a four-tuple (P, T, IN, OUT) where

$P = \{p_1, p_2, p_3, \ldots, p_n\}$ is a set of places,
$T = \{t, t_2, \ldots, t_m\}$ is a set of transitions,
$P \cup T \neq \varnothing$; $P \cap T = \varnothing$;

IN: $(P \times T) \rightarrow N$ is a function called the input function and associates directed arcs from places to transitions, and
OUT: $(P \times T) \rightarrow N$ is an output function that defines directed arcs from transitions to places. ●

Pictorially, places are represented by circles and transitions by horizontal bars. If $IN(p_i, t_j) = k$, where $k \geqslant 1$ is an integer, a directed arc from place p_i to transition t_j is drawn with label k. If $IN(p_i, t_j) = 0$, no arc is drawn from p_i to t_j. Similarly, if $OUT(p_i, t_j) = k$, a directed arc is included from transition t_j to place p_i, with label k if $k > 1$ and without label if $k = 1$. If $k = 0$, no arc is included from t_j to p_i.

DEFINITION 2

Let 2^P be the powerset of P. We then define functions

IP: $T \rightarrow 2^P$ and OP: $T \rightarrow 2^P$ as follows.

$IP(t_j) = \{p_i \in P: IN(p_i, t_j) \neq 0\} \quad \forall t_j \in T$,
$OP(t_j) = \{p_i \in P: OUT(p_i, t_j) \neq 0\} \quad \forall t_j \in T$,

$IP(t_j)$ is called the set of input places of t_j, and
$OP(t_j)$, the set of output places of t_j. ●

DEFINITION 3

A marking M of a Petri net G is a function M: $P \to N$. A marked Petri net W is a Petri net G together with a marking defined on it. We denote it by (G, M) and write $W = (G, M)$. We always associate an initial marking M_0 with a given PN. M_0 will represent the initial state of the system which the PN is modelling. ●

It can be noted that a marking of a Petri net with n places is an n-dimensional vector and associates with each place a certain number of tokens which are represented by means of dots inside the places.

DEFINITION 4

A transition t_j of a Petri net is said to be enabled in a marking M iff

$$M(p_i) \geqslant IN(p_i, t_j) \quad \forall p_i \in IP(t_j).$$

An enabled transition t_j can fire at any instant of time. When a transition t_j, enabled in a marking of M fires, a new marking M' is reached according to the equation,

$$M'(p_i) = M(p_i) + OUT(p_i, t_j) - IN(p_i, t_j) \quad \forall p_i \in P,$$

we say marking M' is reachable from M and write $M \xrightarrow{t_j} M'$. ●

DEFINITION 5

The set of all markings reachable from an initial marking M_0 of a Petri net is called the reachability set of M_0 and is denoted by $R[M_0]$. ●

DEFINITION 6

A GSPN is a six-tuple (P, T, IN, OUT, M_0, F) where

(a) (P, T, IN, OUT, M_0) is a marked Petri net;
(b) T is partitioned into two sets: T_I of immediate transitions and T_T of timed transitions;
(c) F is a function with domain $R[M_0] \times T_T$, which associates to each $t \in T_T$ in each $M \in R[M_0]$, a continuous random variable that indicates the firing time of t in M; and
(d) each $t \in T_I$ has zero firing time in all reachable markings. ●

In the graphical representation of GSPN, a horizontal line represents an immediate transition and a rectangular bar represents a timed transition. GSPN markings are classified into two types: vanishing markings (those in which at least one immediate transition is enabled) and tangible markings (those in which only timed transitions are enabled). In vanishing markings, as a rule, only an immediate transition is selected to fire even if timed transitions are enabled.

In a vanishing marking, if two or more conflicting immediate transitions are enabled, then any of them may fire, according to a predefined probability distribution. The set of conflicting transitions together with associated probabilities is referred to as a

random switch. A random switch may be static or dynamic. In a static random switch, the switching probabilities are constant whereas in a dynamic random switch, the probabilities will depend on the current marking of the GSPN. Also, note that the firing rates of exponential transitions are in general marking dependent. Dynamic random switches and marking dependent firing rates add significantly to the modelling power of GSPN.

Often, special arcs called inhibitor arcs are employed to test for the absence of a token and capture priorities. An inhibitor arc from a place p to a transition t is indicated by a directed arc with a black dot at the end. t will now be enabled only if p has no token and all its other (normal) input places have adequate numbers of tokens.

2.2 *Integrated PFQN-GSPN models*

IQP modelling is based on the concept of flow-equivalence. The basic idea is to isolate one or more subsystems of a given system and compute the flow-equivalents for these subsystems. The overall model (also called the high level model) of the given system will then have these subsystems represented by the flow-equivalents. To compute the flow-equivalent of a subsystem, one has to evaluate a model (PFQN or GSPN) of the subsystem for all possible populations (i.e., number of jobs in the subsystem) and compute the throughput in each case. Thus an IQP model can be of two types: (i) IQP model with GSPN as the high level model and (ii) IQP model with PFQN as the high level model. If the high level model is a GSPN, then some of its transitions represent flow-equivalents of subsystems computed using PFQN models of the subsystem. If the high level model is a PFQN, some of its individual nodes correspond to flow-equivalents of subsystems computed using GSPN models of these subsystems.

It is not necessary that the high level model and the models for computing flow-equivalents be of different types (i.e., one is a PFQN and the other a GSPN). If the high level model and the model for flow-equivalence are both PFQN, then we have the classical flow-equivalence of queueing networks (Chandy *et al* 1975). If one is a PFQN and the other a GSPN, we have the integrated PFQN–GSPN models (IQP models) which are the main focus of this paper.

3. IQP models for central server FMS

The closed central server network model of figure 1 is a well-formulated representation for an FMS with a transport mechanism such as automated guided vehicles (AGV) and several versatile numerically controlled (NC) machines. The network has two subsystems which we shall call the AGV subsystem and the machine subsystem. The transport mechanism serves as the central server and the probabilities $q_0, q_1, \ldots, q_m$ help capture the routing of parts inside the system. The closedness of the network is ensured by the availability of a limited number of fixtures for parts inside the system. Thus the population of jobs inside the network is equal to the total number of fixtures, assuming a perennial supply of raw parts. The central server network is an easily solvable PFQN but in this section, we shall consider variants of this model by including several non-product form features. Also, for ease of discussion, we consider a model with one AGV and two machines M_1 and M_2.

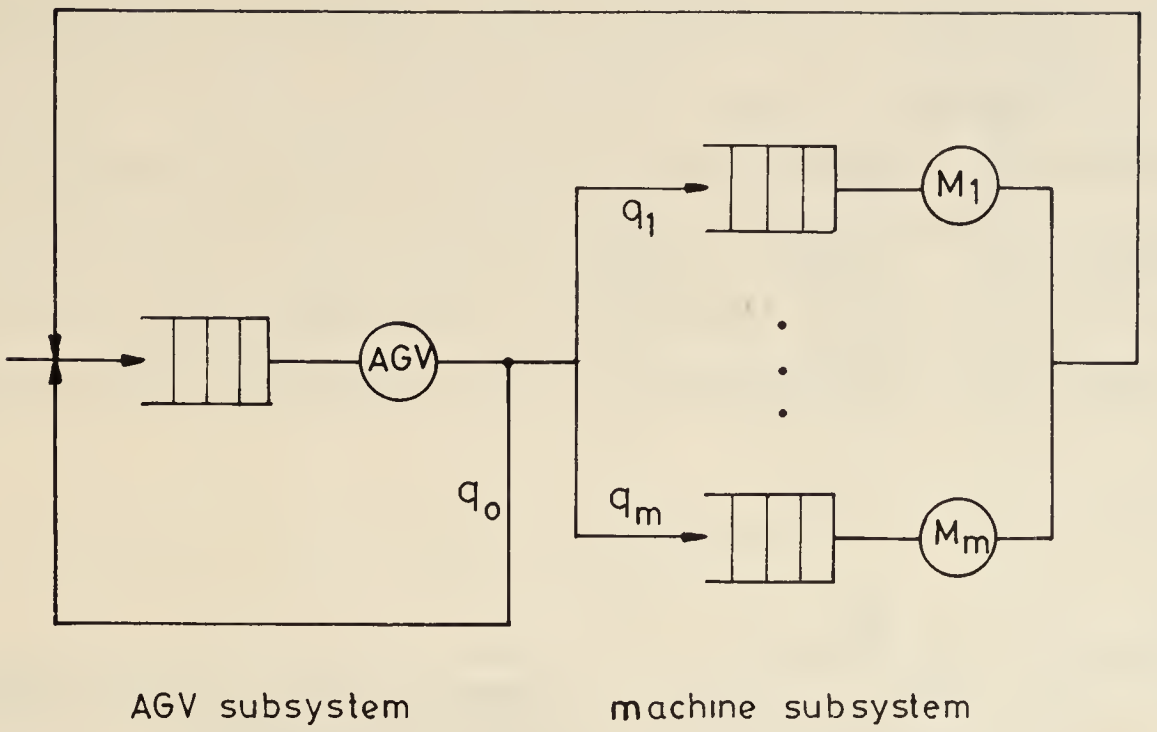


Figure 1. Closed central server model of an FMS with m machines and one AGV.

3.1 *Example* 1: *FMS with dynamic routing*

In this example, we construct an IQP model of a central server FMS with parts routed to the machines M_1 and M_2 dynamically. The dynamic routing policy we consider is the SQR (shortest queue routing) which has a good deal of intuitive appeal. To construct an IQP model, we first observe that the non-product form feature, SQR in this case, is localized in the subsystem comprising the machines M_1 and M_2. Therefore, a GSPN model of this subsystem can be used to compute the flow-equivalent. The AGV node together with this flow-equivalent will make up a high level PFQN model as shown in figure 2. The GSPN model for the subsystem is shown in figure 3, with description of the model in table 1.

In the GSPN model for computing the flow-equivalent, observe that the AGV is short circuited. The transitions t_1 and t_2 will constitute a dynamic random switch that implements SQR routing, as follows.

$$\begin{aligned}
\text{Prob}(t_1) &= 0, && \text{if } M(p_2) > M(p_3),\\
&= 1, && \text{if } M(p_2) < M(p_3),\\
&= 0{\cdot}5, && \text{if } M(p_2) = M(p_3).
\end{aligned}$$

$$\begin{aligned}
\text{Prob}(t_2) &= 0, && \text{if } M(p_2) > M(p_3),\\
&= 1, && \text{if } M(p_2) < M(p_3),\\
&= 0{\cdot}5, && \text{if } M(p_2) = M(p_3).
\end{aligned}$$

Note in the above that ties are resolved with equal probabilities.

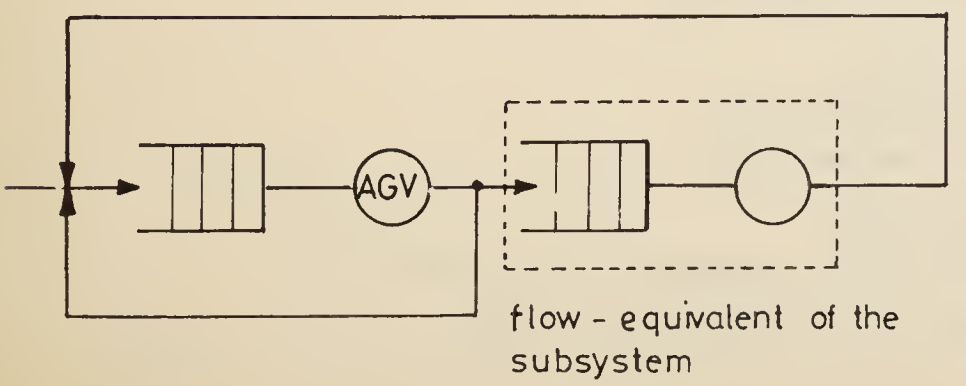


Figure 2. High level PFQN model for example 1.

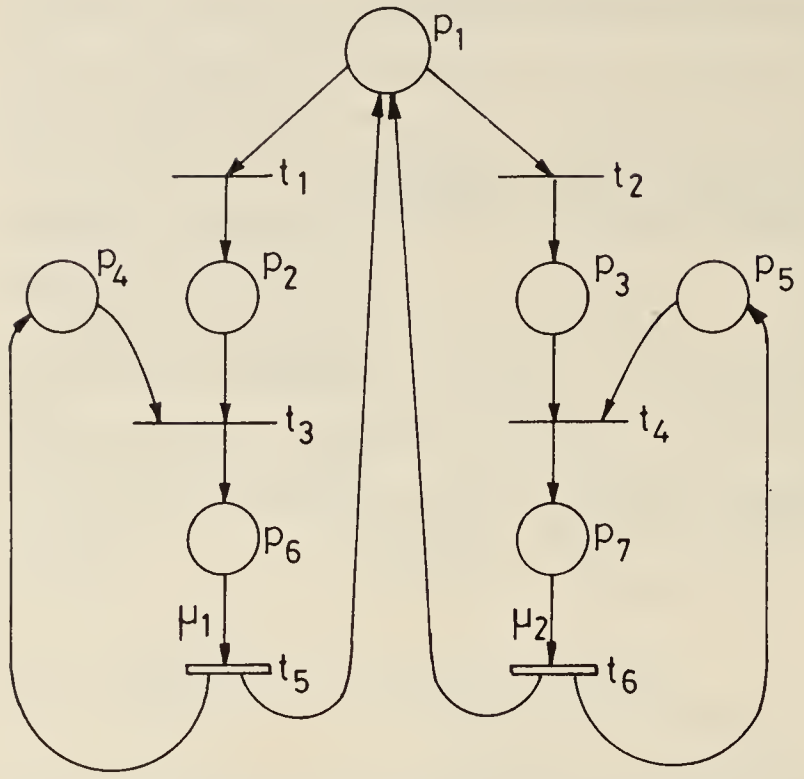


Figure 3. GSPN model for computing the flow-equivalent in example 1.

To compute the flow-equivalent, we observe that the number of jobs in the subsystem at any time can only vary from 0 to N where N is the total number of jobs in the closed network model of our FMS. The GSPN model is to be run for each of the populations 0, 1, 2, , . . . , N and to compute the service rate of the flow-equivalent server, we have to sum the throughput rates of the transitions t_5 and t_6. The PFQN high level model of figure 2 can now be solved using standard computational algorithms. It would now be very efficient to study the effect of variations of parameters connected with the AGV node on the performance of the overall system because the experimentation is required on a 2-node PFQN.

To see the computational advantage obtained, notice that an exact GSPN model of the entire system will have $O(N)$ states where N is the total number of jobs. Solving

Table 1. GSPN model for computing the flow-equivalent in example 1. (See figure 3.)

Places	
1	Parts ready to be routed to M_1 or M_2
2	Queue of parts waiting for M_1
3	Queue of parts waiting for M_2
4	M_1 available
5	M_2 available
6	M_1 processing a part
7	M_2 processing a part
Immediate transitions	
1	Part routed to queue for M_1
2	Part routed to queue for M_2
3	M_1 starts processing a part
4	M_2 starts processing a part
Exponential transitions	
5	Processing of part by M_1
6	Processing of part by M_2
Dynamic random switch	
The transitions t_1 and t_2 constitute a dynamic random switch to model shortest queue routing.	

the exact GSPN model will therefore entail solving a Markov chain with $O(N)$ states. On the other hand, the GSPN submodel of figure 3 has at most two states for any value of N, because of SQR routing. In particular, when the population n is even, the submodel has only one tangible state and for odd n, there are exactly two tangible states. This would mean that the flow-equivalent can be obtained by solving $(N + 1)$ Markov chains of at most 2 states, each. Finally, the solution of the PFQN high level model is also a very trivial affair. We can therefore conclude that the exact GSPN model leads to a time complexity of $O(N^3)$ whereas the IQP model leads to $O(N)$ complexity.

3.2 *Example 2: FMS with finite demand for products*

Here, we employ a GSPN high level model and compute the flow-equivalent using a PFQN submodel. The system considered is the central server FMS with a non-product form feature in the AGV subsystem rather than in the machine subsystem.

The FMS in question, depicted in figure 4, has a total of N fixtures, thus limiting the maximum number of in-process jobs to N. Raw parts are always available, however, a fixture released before the unloading of a finished part is not utilized for a waiting raw part until a specific demand for a finished part is posted in the demand queue. These demands are random and are assumed to constitute a Poisson arrival stream. When k $(k > 0)$ demands are waiting unfulfilled, no further demands are entertained and the demands that arrive when the demand buffer is full are assumed to be lost. There are two potential non-product form features in the AGV subsystem here, namely finiteness of the demands buffer and the joining operation for demands and fixtures. Assuming product-form characteristics for the machine subsystem, we then have a model that is congenial for IQP modelling. Note that the overall model here is not a closed queueing network, but a network in which a maximum of N jobs can circulate.

A high level GSPN model of the above system is shown in figure 5 (description of the model in table 2). In this model, t_7 is an exponential transition that aggregately represents the machine subsystem. Needless to say, t_7 has a marking dependent firing rate that can be computed by solving the PFQN model shown in figure 6.

Thus the solution of the IQP model in this case proceeds in the following way. In stage 1, the flow-equivalent of the PFQN model of figure 6 is computed by solving this simple model for different populations n ranging from 0 to N. This determines

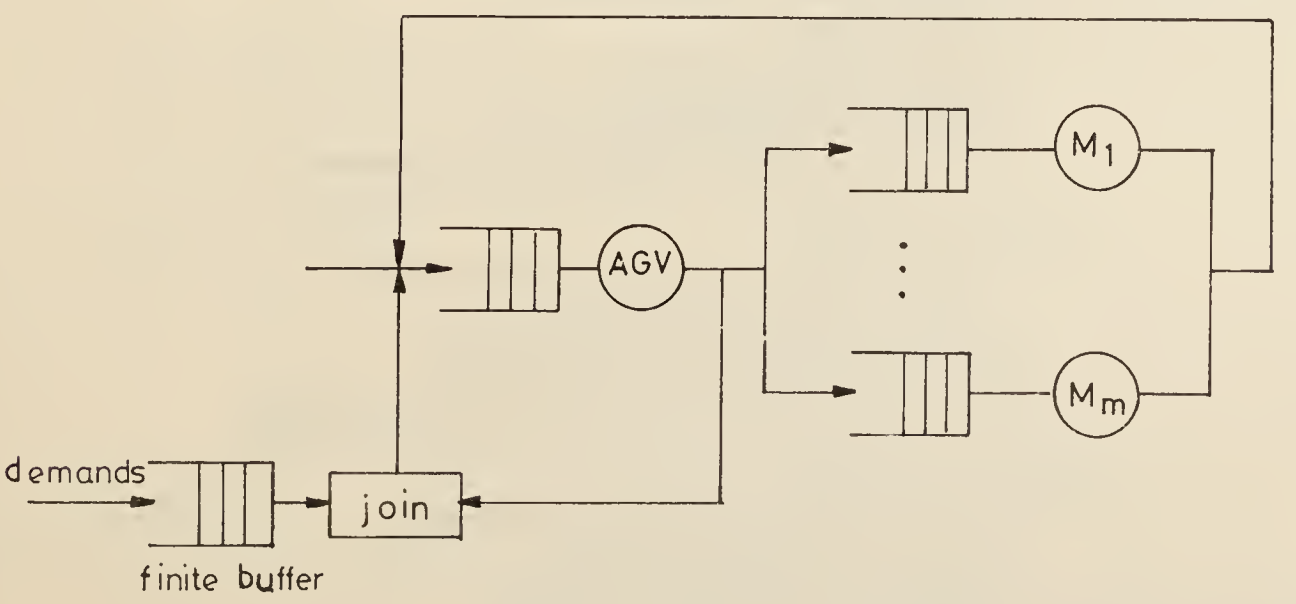


Figure 4. Central server FMS model with a finite external demand for finished products.

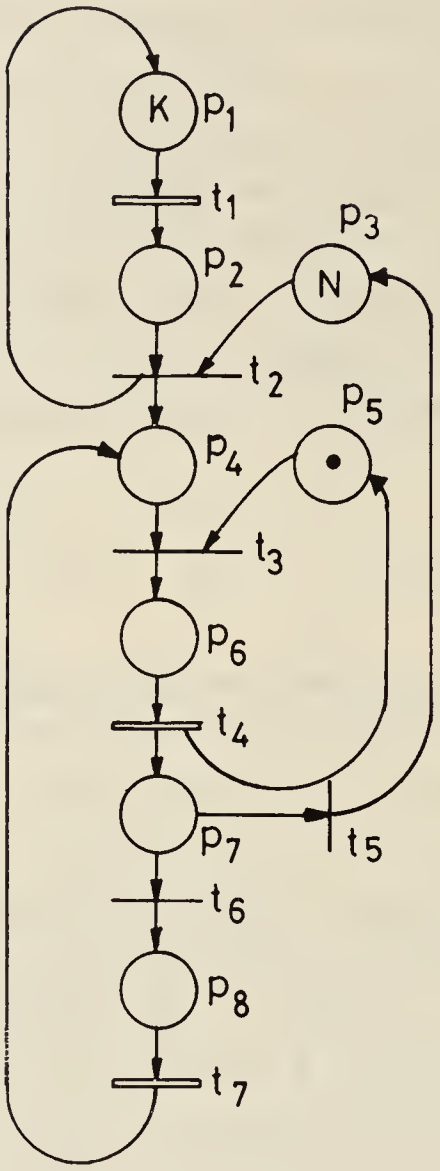


Figure 5. High level GSPN model for example 2.

the marking dependent firing rates of the transition t_7. In stage 2, the GSPN model of figure 5 is analysed using the standard techniques. This GSPN model has only 3 exponential transitions whereas the exact GSPN model for the entire system has $m + 2$ exponential transitions where m is the number of machines.

Table 2. High level GSPN model for example 2. (See figure 5.)

Places	
1	Future demands for products
2	Queue of waiting demands for products
3	Fixtures available
4	Queue of parts waiting for AGV
5	AGV available
6	AGV transporting a part
7	A part just transported by AGV
8	Service in progress in the machine subsystem
Immediate transitions	
2	A fixtured part is released into the system
3	AGV starts transporting a part
5	Part carried by AGV is for unloading
6	Part carried by AGV is for some more machining
Exponential transitions	
1	Arrival of demands for finished products
4	Transport operation by AGV
7	Service in the machine subsystem

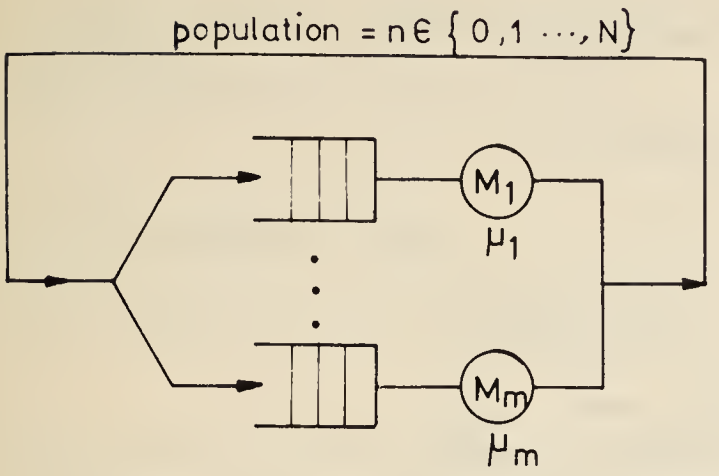


Figure 6. PFQN model for computing the flow-equivalent in example 2.

4. Error analysis

The integrated PFQN–GSPN modelling technique is essentially an approximate technique for obtaining the mean values of performance measures. It has the advantages of computational efficiency of PFQN and representational power of GSPN. The accuracy of performance estimates using the integrated technique will depend on the degree of coupling between the subnetworks. The method leads to highly accurate estimates if this degree of coupling is very small. We shall see this with a simple example abstracted from the central server model we discussed in § 3.

Example 3. Consider a performance model with two subnetworks A and B, as shown in figure 7. Let the network B contain some non-product form features whereas the network A is product form. If we employ the IQP modelling, we typically compute the flow-equivalent of network B using a GSPN model which is run for different populations in the network. The computation of the flow-equivalent assumes that the duration for which a population in network B remains constant is long enough to offset any interference from network A. That is, between two successive interactions between the subnetworks A and B, network B reaches an equilibrium. If this condition is not met, then the flow-equivalent is bound to be inaccurate. In the specific case of the model of figure 6, this condition is met when either the value of p is close to 1 or the relative throughput is much higher than that of A.

We shall now present some numerical experiments conducted with examples 1 and 2, to see the accuracy of the IQP modelling technique. Table 3 shows the mean throughput and mean lead time values for example 1, with the following parameters.

μ_0 = AGV service rate = 50 parts per hour;
$\mu_1 = M_1$ service rate = 10 parts per hour;
$\mu_2 = M_2$ service rate = 5 parts per hour;
$q_1 = q_2 = (1 - q_0)/2$;
q_0 varies from 0·01 to 0·99.

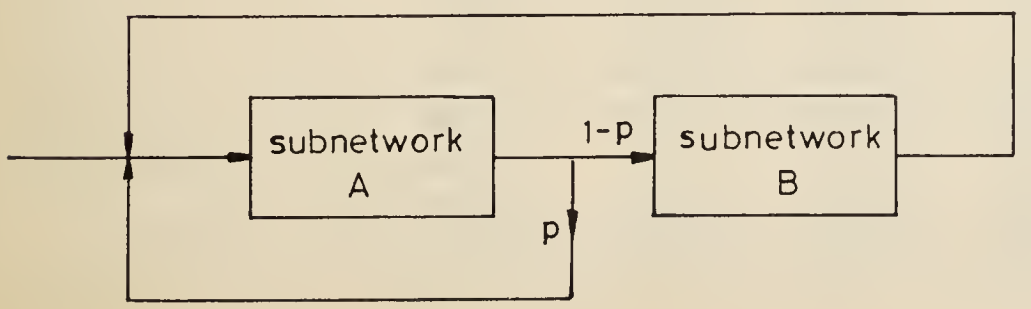


Figure 7. Performance model for example 3.

Table 3. Mean throughput and mean lead time for example 1.

$\mu_0 = 50$ parts per hour; $\mu_1 = 10$ parts per hour; $\mu_2 = 5$ parts per hour; $q_1 = q_2 = (1 - q_0)/2$;
Number of fixtures = 8.

	Mean throughput		Mean lead time	
q_0	Exact GSPN model	IQP model	Exact GSPN model	IQP model
0·99	49·50	49·50	0·16161	0·16161
0·95	47·49992	47·49993	0·16842	0·16842
0·90	44·98879	44·98999	0·17782	0·17781
0·80	39·14079	39·21808	0·20439	0·20398
0·70	30·37088	30·60465	0·26341	0·26139
0·60	21·44702	21·63114	0·37301	0·36983
0·50	14·71864	14·81111	0·54352	0·54013
0·40	9·91265	9·95348	0·80705	0·80374
0·30	6·39908	6·41611	1·25017	1·24686
0·20	3·74009	3·74664	2·13898	2·13524
0·10	1·66394	1·66592	4·80784	4·80215
0·05	0·78844	0·78921	10·14656	10·13663
0·01	0·15134	0·15147	52·85555	52·85521

Note that the throughput is the number of finished parts per hour and the lead time is the average time spent by a job inside the system. The numerical results in table 3 show the estimates using the IQP modelling and exact GSPN modelling. The estimates obtained through IQP models are found to be remarkably accurate, specially for small values of q_0.

Table 4. Mean throughput and mean lead time for example 2.

$\mu_0 = 50$ parts per hour; $\mu_1 = 10$ parts per hour; $\mu_2 = 5$ parts per hour; $q_1 = q_2 = (1 - q_0)/2$;
Number of fixtures = 4.

	Mean throughput		Mean lead time	
q_0	Exact GSPN model	IQP model	Exact GSPN model	IQP model
0·99	19·86499	19·86499	0·03396	0·03396
0·90	19·62480	19·62429	0·05668	0·05671
0·80	18·50368	18·49833	0·09763	0·09784
0·70	15·84976	15·83368	0·16315	0·16376
0·60	12·29868	12·27539	0·26070	0·26186
0·50	8·92581	8·90384	0·40232	0·40405
0·40	6·17814	6·16185	0·61550	0·61781
0·30	4·04035	4·03002	0·96859	0·97164
0·20	2·37657	2·37095	1·67004	1·67440
0·10	1·06081	1·05850	3·76461	3·77304
0·05	0·50316	0·50211	7·94671	7·96081
0·01	0·09664	0·09645	41·38727	41·47170

Table 4 shows similar results for example 2. The accuracy of IQP modelling is again illustrated by these results.

5. Concluding remarks

5.1 *Advantages and limitations*

In this paper, we have shown through illustrative examples, that integrated PFQN–GSPN models provide an efficient and accurate performance modelling paradigm for flexible manufacturing systems. We collect together below the main advantages of the IQP models.

(1) IQP models have a solution efficiency comparable to that of PFQN and a representational power equalling that of GSPN.
(2) The flow-equivalents computed in the integrated technique are exact since they are computed by solving the PFQN or GSPN.
(3) The accuracy of solution given by IQP models is very good, especially when the subsystems in the given system are loosely coupled.
(4) In the area of queueing networks, different kinds of approximations have been developed for different non-product form features. The integrated models provide a unified technique for all non-product form features. Also, flow-equivalents computed using queueing networks are approximate and not exact.
(5) When two or more non-product form features are present in a subsystem, there are few techniques available in queueing theory to handle them. The IQP technique can handle any combination and any number of non-product features in a subsystem, in a unified way.

A limitation of the integrated technique is that it could be used only to compute average performance measures. However, this is true of any approximation technique based on flow-equivalence. Another limitation arises when the non-product form features are distributed across all subsystems. In such a case, the efficiency of IQP models suffers. The efficiency of these models will be appealing if the non-product form features are localized in a single subsystem.

5.2 *Use of other modelling paradigms*

In this paper, we have specifically looked at integrated models where PFQN and GSPN are employed. It is needless to add here that other analytical models could be used in the place of the above two models. For example, the flow-equivalent for the SQR subsystem in example 1, which was computed using a GSPN model (figure 3), can as well be computed by visualizing the underlying Markov chain model and solving it. In fact, in this particular case, it is easy to write down the generator matrix of the underlying Markov chain in a straightforward way. In a general situation where it is hard to visualize let alone compute the generator matrix of the underlying Markov chain, GSPN provide an easy-to-use, high level paradigm for automatic computation of the Markov model.

As another example, let us consider the central server FMS model of figure 4 and assume that the routing is SQR in the machine subsystem. In this situation, both the

AGV subsystem and the machine subsystem have a non-product form feature. Three ways of solving it become immediately obvious: (1) Exact GSPN modelling; (2) compute flow-equivalent of AGV subsystem using GSPN and then solve a high level GSPN model; (3) compute flow-equivalent of machine subsystem using GSPN and then solve a high level GSPN model. In methods 2 and 3, all the models are GSPN. For numerical results using methods 2 and 3, see Hanumantha Rao (1991). If one wishes, one could use any combination of analytical or simulation models.

References

Balbo G, Bruell S C, Ghanta S 1986 Combining queueing networks and generalized stochastic Petri nets for the analysis of software blocking phenomena. *IEEE Trans. Software Eng.* SE-12: 561–576

Buzacott J A, Yao D D 1986 On queueing network models of flexible manufacturing systems. *Queueing Systems* 1: 5–27

Chandy K M, Herzog V, Woo L 1975 Parametric analysis of queueing networks. *IBM J. Res. Dev.* 19: 36–42

Chandy K M, Sauer C H 1978 Approximate methods for analysis of queueing network models of computer systems. *ACM Comput. Surv.* 10: 263–280

Hanumantha Rao P 1991 *Integrated analytical models for automated manufacturing systems*, M E Project Report, Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore

Ho Y C 1987 Performance evaluation and perturbation analysis of discrete event systems. *IEEE Trans. Autom. Control* AC-32: 563–572

Meenakshisundaram C R 1990 *Integrated analytical models for parallel and distributed computing systems*, M Sc (Eng.) thesis, Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore

Murata T 1989 Petri nets: Properties, analysis, and applications. *Proc. IEEE* 77: 541–580

Ramesh N R 1990 *Integrated performance models based on queueing networks and Petri nets*, M E Project Report, Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore

Ranky P G 1983 *The design and operation of flexible manufacturing systems* (Amsterdam: IFS and North-Holland)

Viswanadham N, Narahari Y 1988 Stochastic Petri nets for performance evaluation of automated manufacturing systems. *Inf. Decision Technol.* 14: 125–142

Markovian models for deadlock analysis in automated manufacturing systems

Y NARAHARI, N VISWANADHAM and K R KRISHNA PRASAD

Department of Computer Science & Automation, Indian Institute of Science, Bangalore 560012, India

Abstract. Deadlocks constitute a major issue in the design and operation of discrete event systems. In automated manufacturing systems, deadlocks assume even greater importance in view of the automated operation. In this paper, we show that Markov chains with absorbing states provide a natural model of manufacturing systems with deadlocks. With illustrative examples, we show that performance indices such as mean time to deadlock and mean number of finished parts before deadlock can be efficiently computed in the modelling framework of Markov chains with absorbing states. We also show that the distribution of time to deadlock can be computed by conducting a transient analysis of the Markov chain model.

Keywords. Markovian models; deadlock analysis; automated manufacturing systems; discrete event systems; mean time to deadlock.

1. Introduction

A typical automated manufacturing system (AMS) comprises a computer controlled configuration of numerically controlled (NC) machine tools and a material handling system (MHS) designed to simultaneously manufacture low to medium volumes of high quality products at low cost. The architecture of such a system is constituted by

(a) versatile NC machines which can carry out a variety of machining operations,
(b) an automated MHS comprising conveyors, or automated guided vehicles (AGV) to move parts and tools between machines,
(c) a hierarchical control system that coordinates the actions of the machines, the MHS, and the workpieces,
(d) a load/unload workstation through which the entry and exit of a part occurs; a fixturing station where the parts entering the system are fixtured onto pallets; inspection stations; and coordinate measurement machines, and
(e) a buffer storage in the form of local storage or central storage or both, to store raw and semi-finished workpieces.

An AMS can be described as a discrete event dynamical system (Ho 1987) because changes in the system state are caused by the occurrence of events at discrete instants of time instead of continuously. Therefore the evolution of states in an AMS is not

completely described by partial or ordinary differential equations, as in the case of continuous variable dynamical systems. Examples of discrete events in an AMS are: entry/exit of a part, starting/finishing of part transfer by a robot, starting/finishing of processing by a machine, a robot failure, and a machine breakdown. The number of such activities in a typical AMS is large and there are numerous interactions involving these activities. Further, these interactions exhibit concurrency, contention for resources, synchronization, and randomness. The interactions might also lead to a deadlocked state, in which the system is crippled and produces no output. Any effective model of an AMS should therefore capture the above characteristics.

The need for and the role of mathematical models in automated manufacturing systems has been clearly brought out by Suri (1985) and by Buzacott & Yao (1986). AMS models can be used in a variety of ways such as in logical analysis, performance prediction, optimization, control, and for gaining an insight into the system.

There are two basic types of models of discrete event dynamical systems and hence AMS: qualitative models and quantitative models. Qualitative models such as Petri nets (Narahari & Viswanadham 1985), finite state machines (Wonham 1987), and general algebraic discrete event models (Inan & Varaiya 1989) are useful for investigating logical aspects of FMS behaviour such as boundedness, fairness, absence/existence of deadlocks, mutual exclusion of resources, and correctness of control logic. Quantitative models are basically stochastic models that address quantitative issues of performance such as throughput, manufacturing lead time, machine/robot/AGV utilizations, and reliability. There are three basic types of quantitative modelling tools: simulation models, analytical models, and hybrid models, which use both simulation and analytical models. In this paper, we are concerned with analytical models of automated manufacturing systems.

Markov chains constitute the basic stochastic model of automated manufacturing systems. For real-life systems, Markov chain models have a large state space, entailing large scale computation for transient or steady-state analysis. Further, it is often difficult to generate the Markov chain model directly from the system. Queueing network (QN) models (Buzacott & Yao 1986) effectively overcome the state space explosion problem inherent in Markovian modelling. QN models capture the dynamics, interactions, and uncertainties in the system in an aggregate way and the performance measures produced are average values which assume a steady-state operation of the system. The computational efficiency of QN models applies only to a special class of QN called product form queueing networks. For non-product form systems, exact modelling can only be done via explicit Markov chain analysis. The generation of a Markov chain model can be completely automated with the help of generalized stochastic Petri nets (GSPN) (Marsan *et al* 1986). GSPN constitute a high level modelling tool which can elegantly capture the characteristics of discrete event dynamical systems.

In this paper, we show that the framework of Markov chains with absorbing states can be used to model AMS with deadlocks.

2. Deadlocks in automated manufacturing systems

Deadlocks constitute an undesirable phenomenon in AMS. A deadlock is a situation where each of a set of two or more parts keeps waiting indefinitely for the other parts in the set to release resources. In the context of AMS, resources refer to machines,

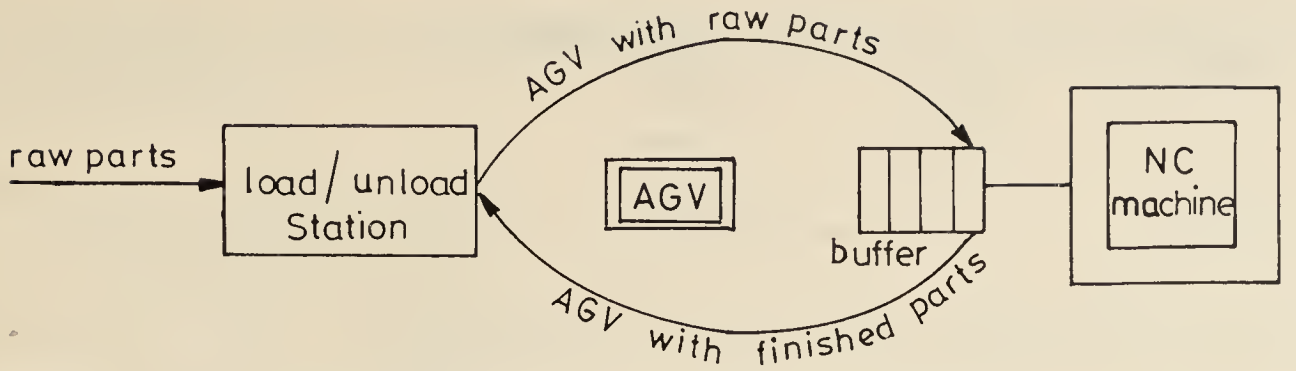


Figure 1. A simple AMS to illustrate deadlock.

buffers, AGV, robots, fixtures, tools etc. Deadlocks may arise as the final state of a complex sequence of operations on several parts passing concurrently through a system, and thus are generally difficult to predict. Deadlocks lead to degraded performance and ultimately zero throughput. The automated operation of an AMS makes deadlocks an important problem in the design of these systems. In an improperly designed AMS, the only remedy for a deadlock may be manual clearing of machines or AGV or buffers, and restart of the system from an initial condition which is known to produce deadlock-free operation under nominal production conditions. The loss of production and the labour cost in resetting the system in this way can be avoided by proper design. Often, deadlocks may result due to simple software bugs in the controllers.

Example 1. To visualize a deadlock, consider the simple AMS depicted in figure 1. There is a load/unload (L/U) station at which raw parts arrive randomly. An AGV carries a raw part from the L/U station to an NC machine which carries out some operations on the raw part. The finished part is carried by the AGV to the L/U station where the finished part is unloaded. After depositing a raw part on the NC machine, the AGV travels back to the L/U station. It will pick up a raw part if one is available and travel to the NC machine. It is assumed that the AGV can only carry one part at a time. Also the AGV takes a negligible amount of time to travel from the L/U station to the machine and vice versa. Imagine the following sequence of events, starting with an initial state in which the AGV and the machine are free and raw parts are available: (i) The AGV carries a raw part, say part 1, and loads it onto the NC machine, which starts processing part 1. (ii) The AGV returns to the L/U station. Before the machine finishes processing, another raw part arrives and the AGV carries the raw part, say part 2, to the machine, but waits for the machine which is still processing part 1. Thus the AGV gets blocked, waiting for the machine. (iii) The machine finishes the operations on part 1 and starts waiting for the AGV to carry the finished part 1 to the L/U station. At this juncture, the machine gets blocked waiting for the AGV. If the machine and the AGV can only accommodate one part at a time and there is no additional buffer space, then the two resources here are involved in a deadlock since each keeps waiting for the other indefinitely. Figure 2 gives a pictorial description of the above sequence of events. Even if some buffer space is provided for raw parts and finished parts in the above system, a deadlock can still occur because the AGV can fill the entire buffer with raw parts during the processing of part 1 by the machine.

Important performance issues in AMS with deadlocks would be: Mean time before deadlock, mean throughput before deadlock, probability of a particular deadlock, and so on. In this paper, we show that a natural model for deadlocks is provided by the absorbing states of a Markov chain. The above performance issues can be effectively addressed using the theory of Markov chains with absorbing states (Trivedi 1982).

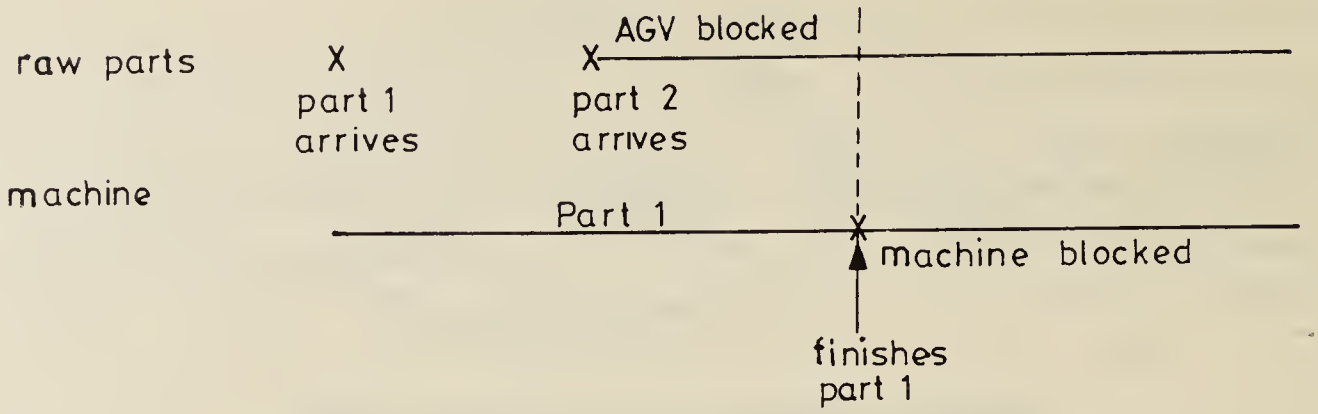


Figure 2. Sequence of events leading to deadlock.

3. Modelling of deadlocks by absorbing states

Here, we present the relevant theory of Markov chains with absorbing states (Trivedi 1982) and show with an example, the application to performance evaluation in the presence of deadlocks.

Let the state space of a finite discrete time Markov chain (DTMC) $\{X_n: n = 0, 1, 2, \ldots\}$ be given by

$$S = \{1, 2, \ldots, m, m+1, m+2, \ldots, m+n\},$$

where $m > 0$, $n > 0$, the first m states are transient states, and the rest of the states are absorbing states. The transition probability matrix (TPM) of the above DTMC may be partitioned as

$$P = \left[\begin{array}{c|c} T & C \\ \hline \mathbf{0} & I \end{array}\right].$$

In the above, T is an $(m \times m)$ matrix that describes single step transitions among transient states; I is the identity matrix of order n; $\mathbf{0}$ is an $(n \times m)$ zero matrix; and C is an $(m \times n)$ matrix that describes transitions from transient states into absorbing states. The k-step TPM can be seen to be

$$P^k = \left[\begin{array}{c|c} T(k) & C(k) \\ \hline \mathbf{0} & I \end{array}\right],$$

where

$$T(k) = T^k$$

and

$$C(k) = (T^{k-1} + T^{k-2} + \cdots T + I)C.$$

Note that $[T^k]_{ij}$, $1 \leqslant i, j \leqslant m$, is the probability of arriving in transient state j, starting from transient state i, after exactly k steps.

It can be shown that the $(\lim_{t \to \infty} \Sigma_{k=0}^{t} T^k)$ exists and is given by the matrix $F = (I - T)^{-1}$.

F is called the Fundamental Matrix and is a rich source of information about the given Markov chain with absorbing states. It can be shown (Trivedi 1982) that the (i,j)th element of the fundamental matrix gives the mean number of times transient state j is visited, starting from transient state i, before the Markov chain reaches any absorbing state. Note that

$$F = I + F\,T.$$

Thus the matrix F can be computed by solving the above system of linear simultaneous equations.

3.1 *Mean time to deadlock*

Let state 1 be the initial (transient) state. Then for $j = 1, 2, \ldots, m, f_{1j}$ would give the mean number of visits to state j before deadlock. If m_j is the mean sojourn time in state j, then the mean time to deadlock

$$= \sum_{j=1}^{m} f_{1j} m_j.$$

More generally, let transient state j be the initial state with probability p_j, where

$$\sum_{j=1}^{m} p_j = 1.$$

Then, the mean number of visits to state j before deadlock

$$= \sum_{i=1}^{m} p_i f_{ij}.$$

In this case, the mean time to deadlock

$$= \sum_{j=1}^{m} \left(\sum_{i=1}^{m} p_i f_{ij} \right) m_j.$$

Note that the above results would hold for a continuous time Markov chain, by interpreting the above DTMC as an embedded Markov chain.

3.2 *Probability of absorbing states*

Let us look at the matrices

$$C(k) = (T^{k-1} + T^{k-2} + \cdots + T + I)C, \text{ and } G = FC.$$

The (i,j)th element of the matrix $C(k)$, $1 \leqslant i \leqslant m$, $1 \leqslant j \leqslant n$, gives the probability that the Markov chain would reach the absorbing state $m + j$, starting from transient state i, in exactly k steps. g_{ij}, the (i,j)th element of G, can be interpreted as the long term probability that the Markov chain reaches the absorbing state $m + j$, starting from transient state i.

Therefore, if state 1 is the initial state, then $g_{11}, g_{12}, \ldots, g_{1n}$ would give the long term probabilities of the Markov chain ending up in the absorbing states $m + 1$, $m + 2, \ldots$, and $m + n$, respectively.

On the other hand, if $p_1, p_2, \ldots$, and p_m are the probabilities that the initial state is the state $1, 2, \ldots$, and m respectively, then, for $j = 1, 2, \ldots, n$, the probability of ending up in absorbing state $(m + j)$ is given by

$$\sum_{i=1}^{m} p_i g_{ij}.$$

The analysis of a continuous time Markov chain (CTMC) with absorbing states will

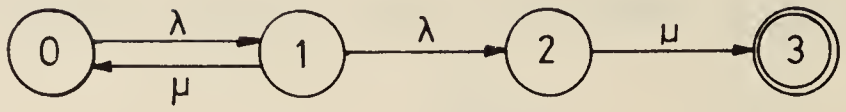


Figure 3. Markov chain model for example 2.

proceed in the same way, except that we have to work with the embedded Markov chain.

Example 2. We now study the system of example 1. Let the raw parts arrive according to a Poisson process with rate λ and let the processing time be exponentially distributed with rate μ. Assume that there is no buffer in front of the NC machine. Then the above system can be modelled as a CTMC with four states $\{1,2,3,4\}$ having the following interpretation.

1: Machine idle, AGV idle, waiting for next raw part;
2: machine processing a part, no raw parts waiting;
3: machine processing a part, AGV waiting at the machine with a raw part;
4: machine, with a finished part, waiting for the AGV, and the AGV, with a raw part, waiting for the machine.

The state transition diagram of the above CTMC is shown in figure 3. If there is a buffer with capacity $n(n>0)$ in front of the machine, then the state space is $\{1,2,3,\ldots, n+3, n+4\}$, with proper interpretation. The state transition diagram of this CTMC is shown in figure 4. Note that there is exactly one absorbing state, namely $n+4$, and the rest are transient states.

The TPM of the embedded Markov chain of the CTMC of figure 3 is given by

$$P=\left[\begin{array}{ccc:c} 0 & 1 & 0 & 0 \\ p & 0 & 1-p & 0 \\ 0 & 0 & 0 & 1 \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right],$$

where

$$p=\mu/(\lambda+\mu).$$

Here

$$T=\begin{bmatrix} 0 & 1 & 0 \\ p & 0 & 1-p \\ 0 & 0 & 0 \end{bmatrix} \text{ and } C=\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

The fundamental matrix can be computed to be given by

$$F=\begin{bmatrix} \dfrac{1}{1-p} & \dfrac{1}{1-p} & 1 \\ \dfrac{1}{1-p} & \dfrac{1}{1-p} & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

Thus, state 1 and state 2 are each visited $1/(1-p)$ times, on an average, before the occurrence of a deadlock. That is, on an average, $1/(1-p)$ parts are produced by the system before the deadlock. If $\mu=8$ parts/hour and $\lambda=2$ parts per hour then $p=0{\cdot}8$ and hence on the average 5 parts are produced before deadlock.

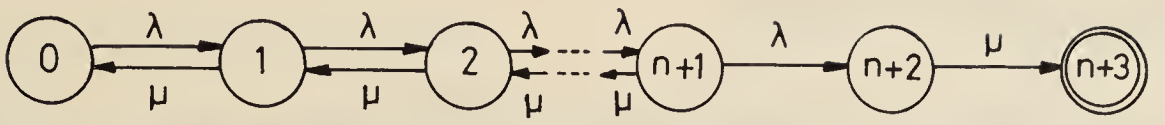


Figure 4. Markov chain model, with a buffer of capacity n.

The mean sojourn times, $m_j (j = 0, 1, 2)$, in the states 0, 1, and 2 can be seen to be

$$m_0 = 1/\lambda; \quad m_1 = 1/(\lambda + \mu); \text{ and } m_2 = 1/\mu.$$

Therefore, the mean time to deadlock

$$= \left(\frac{1}{1-p}\right)\left(\frac{1}{\lambda}\right) + \left(\frac{1}{1-p}\right)\left(\frac{1}{\lambda+\mu}\right) + 1\left(\frac{1}{\mu}\right)$$

$$= 3{\cdot}125 \text{ hours.}$$

From this we can compute the utilization of the machine. If the machine is utilized 100%, then the throughput rate is 8 parts/hour. However, since only 5 parts are produced, the machine is utilized for a mean total of 5/8 hours out of 3·125 hours. Hence machine utilization

$$= \frac{5}{8(3{\cdot}125)} = \frac{5}{25} = \frac{1}{5}.$$

If we want to increase the mean time to deadlock, thereby increasing the number of parts produced before deadlock, then we can introduce a buffer where raw parts and/or finished parts are deposited. The resulting CTMC has been shown in figure 4. It will be interesting to solve this CTMC for different values of n and to notice that the mean time to deadlock is an increasing function of the capacity of the buffer. Figure 5 shows a graph of the mean time to deadlock versus the capacity of the buffer.

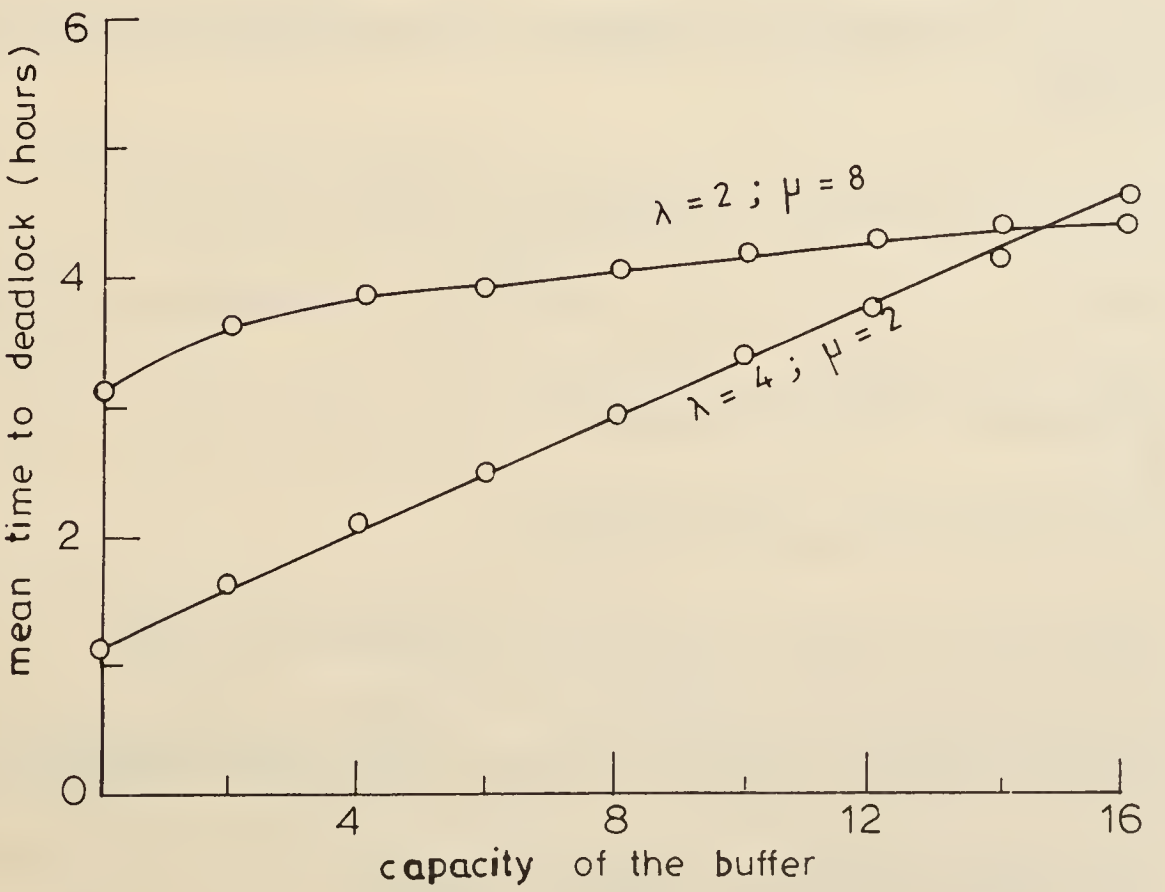


Figure 5. Variation of mean time to deadlock with buffer capacity.

4. Distribution of time to deadlock

In the previous section, it has been shown how the mean time to deadlock can be computed from the fundamental matrix of the Markov chain. If we are interested in the distributions of the time to deadlock, then we need to conduct a transient analysis of the Markov chain model. Let us again consider a CTMC with absorbing states $\{X(t)\colon t \geqslant 0\}$, with state space $\{1, 2, \ldots m, m+1, \ldots, m+n\}$, as in the previous section. Let T be the time to deadlock.

Let

$$p_{ij}(t) = P\{X(t) = j | X(0) = i\}.$$

Then assuming 1 as the initial state, we have

$$P\{T > t\} = P\{X(t) \neq m + j, \text{ for any } j = 1, 2, \ldots, n\},$$

$$= 1 - \sum_{j=1}^{n} p_{1,m+j}(t).$$

Hence the distribution function of T is given by

$$F_T(t) = \sum_{j=1}^{n} p_{1,m+j}(t).$$

The individual probabilities $p_{1,m+j}(t)$ for each j have to be computed by solving the Chapman–Kolmogorov differential equations (Ross 1985). We show the computation of $F_T(t)$ for the system of example 1, by employing Laplace transforms. Such direct methods however do not work for large and complex systems, in which numerical techniques such as uniformization (Gross & Miller 1983) need to be employed.

Example 3. For the Markov chain of figure 3, note that the time T to deadlock has the distribution

$$F_T(t) = p_{14}(t).$$

To compute $p_{14}(t)$, we consider the backward Chapman–Kolmogorov equation for $p_{14}(t)$ (Ross 1985),

$$\frac{\mathrm{d}}{\mathrm{d}t}(p_{14}(t)) = q_{11}p_{14}(t) + q_{12}p_{24}(t) + q_{13}p_{34}(t) + q_{14}p_{44}(t).$$

We know that $q_{11} = -\lambda$; $q_{12} = \lambda$; $q_{13} = 0$; and $q_{14} = 0$. Therefore we have

$$\frac{\mathrm{d}}{\mathrm{d}t}(p_{14}(t)) = -\lambda p_{14}(t) + \lambda p_{24}(t). \tag{1}$$

The backward equation for $p_{24}(t)$ is given by

$$\frac{\mathrm{d}}{\mathrm{d}t}(p_{24}(t)) = q_{21}p_{14}(t) + q_{22}p_{24}(t) + q_{23}p_{34}(t) + q_{24}p_{44}(t).$$

Since $q_{21} = \mu$; $q_{23} = \lambda$; $q_{22} = -(\lambda + \mu)$; and $q_{24} = 0$, the above simplifies to

$$\frac{\mathrm{d}}{\mathrm{d}t}(p_{24}(t)) = \mu p_{14}(t) - (\lambda + \mu)p_{24}(t) + \lambda p_{34}(t). \tag{2}$$

The backward equation for $p_{34}(t)$ is

$$\frac{\mathrm{d}}{\mathrm{d}t}(p_{34}(t)) = -\mu p_{34}(t) + \mu p_{44}(t) = -\mu p_{34}(t) + \mu. \tag{3}$$

We shall solve for $p_{14}(t)$ by the Laplace transform method. Let $P_{ij}(s)$ denote the Laplace transform of $p_{ij}(t)$. Taking the Laplace transform on both sides of (1), (2), and (3), we get respectively

$$sP_{14}(s) = -\lambda P_{14}(s) + \lambda P_{24}(s), \tag{4}$$

$$sP_{24}(s) = \mu P_{14}(s) - (\lambda + \mu)P_{24}(s) + \lambda P_{34}(s), \tag{5}$$

$$sP_{34}(s) = -\mu P_{34}(s) + \mu/s. \tag{6}$$

Simplifying using (4), (5), and (6), we get

$$P_{14}(s) = (\lambda^2 \mu)/\{s(s+\mu)(s^2 + s(2\lambda + \mu) + \lambda^2)\}, \tag{7}$$

which can further be simplified as

$$P_{14}(s) = k/\{s(s+\mu)(s+a)(s+b)\},$$

where the constants k, a, and b depend on λ and μ and can be evaluated easily. Note the pole at $s = 0$ in $P_{14}(s)$. It is to be noted that we are employing here the Laplace transform and not the Laplace–Stieltjes transform. The cumulative distribution function (CDF) of the time to deadlock can now be obtained by taking the inverse Laplace transform. The mean time to deadlock can be obtained by simply differentiating $P_{14}(s)$ and evaluating at $s = 0$. Thus, the solution can be written as

$$p_{14}(t) = A + Be^{-\mu t} + Ce^{-at} + De^{-bt},$$

where A, B, C, and D are constants. Since we know that

$$\begin{aligned} p_{14}(t) &= 0, \quad \text{at } t = 0, \\ &= 1, \quad \text{at } t = \infty, \end{aligned}$$

we have $A = 1$ and $A + B + C + D = 0$.
Thus,

$$p_{14}(t) = 1 + Be^{-\mu t} + Ce^{-at} + De^{-bt},$$

where $B + C + D = -1$.

The above gives the CDF of the time to deadlock. The mean of this random variable has already been shown to be equal to

$$\left(\frac{1}{1-p}\right)\left(\frac{1}{\lambda}\right) + \left(\frac{1}{1-p}\right)\left(\frac{1}{\lambda+\mu}\right) + \frac{1}{\mu},$$

where $p = \mu/(\lambda + \mu)$. Thus the mean time to deadlock can be obtained easily either by differentiating the Laplace transform or by using the fundamental matrix. It is however not so easy to obtain the CDF of the time to deadlock. In general, only transient analysis can help us compute the CDF of the time to deadlock.

5. Concluding remarks

In this paper, we have shown that performance analysis of discrete event systems with deadlocks can be efficiently carried out in the modelling framework of Markov chains with absorbing states. Specifically, we have shown that: (1) mean values of performance measures such as time to deadlock can be computed using the fundamental matrix of the Markov chain, and (2) distributions of these performance measures can be computed by conducting a transient analysis of the Markov chain. A simple illustrative example has been provided to demonstrate the computation of these performance measures. Other interesting examples have been worked out in Krishna Prasad (1990) and Manjunath (1991).

5.1 *Computational issues*

From the computational viewpoint, the computation of mean performance measures entails solving a system of linear simultaneous equations for obtaining the fundamental matrix. To compute the distributions via transient analysis, the Laplace transform method, illustrated in example 3, is feasible only for small sized Markov chains. For real-life FMS, the following numerical methods can be used.

(1) Numerical computation of the matrix exponential;
(2) Runge–Kutta method for solving ordinary differential equations;
(3) uniformization.

There are several methods for evaluating the matrix exponential series (Moler & Van Loan 1978). However, for large Markov chains, these methods are beset with numerical instabilities such as round-off errors. For large Markov chains, the techniques based on the fourth–fifth order Runge–Kutta–Fehlberg method (RKF45 method) (Reibman & Trivedi 1988) and the uniformization or randomization method (Gross & Miller 1983) have proved to be quite efficient. Reibman & Trivedi (1988) have discussed the relative merits of these two methods.

For real-life automated manufacturing systems, another problem to be tackled is the generation of a complete and correct Markov model of the given system. Generalized stochastic Petri nets (Viswanadhan & Narahari 1988) provide a convenient high-level modelling framework for automatically generating the underlying Markov chain. Generalized stochastic Petri nets also provide a convenient framework for detecting deadlocks (absorbing states) in the physical system (Markov model), since the first step in the GSPN analysis is to generate all the states of the underlying Markov chain.

References

Buzacott J A, Yao D D 1986 Flexible manufacturing systems: A review of analytical models. *Manage. Sci.* 32: 890–905

Gross D, Miller D R 1983 The randomization technique as a modelling tool and solution procedure for transient Markov processes. *Oper. Res.* 32: 343–361

Ho Y C 1987 Performance evaluation and perturbation analysis of discrete event systems. *IEEE Trans. Autom. Control* AC32: 563–572

Inan K M, Varaiya P P 1989 Algebras of discrete event models. *Proc. IEEE* 77: 24–38

Krishna Prasad K R *Performance analysis of manufacturing systems with deadlocks*, B E Project Report, Dept. of Computer Science and Automation, Indian Institute of Science, April 1990

Manjunath M S 1991 *Transient analysis of flexible Manufacturing Systems*, M E Project Report, Dept. of Computer Science and Automation, Indian Institute of Science, Bangalore

Marsan M A, Balbo G, Conte G 1986 *Performance models of multiprocessor systems* (Cambridge, Mass: The MIT press)

Moler C, Van Loan C F 1978 Nineteen dubious ways to compute the exponential of a matrix. *SIAM Rev.* 20: 801–835

Narahari Y, Viswanadham N 1985 A Petri net approach to modelling and analysis of flexible manufacturing systems. *Ann. Oper. Res.* 3: 449–472

Reibman A, Trivedi K S 1988 Numerical transient analysis of Markov models. *Comput. Oper. Res.* 15: 19–36

Ross S M 1985 *Introduction to probability models* 3rd edn (Orlando, Fl: Academic Press)

Suri R 1985 An overview of evaluative models for flexible manufacturing systems. *Ann. Oper. Res.* 3: 13–21

Trivedi K S 1982 *Probability and statistics with reliability, queueing and computer science applications* (Englewood Cliffs, NJ: Prentice-Hall)

Viswanadham N, Narahari Y 1988 Stochastic Petri net models for performance evaluation of automated manufacturing systems. *Inf. Decision Technol.* 14: 125–142

Wonham W M 1987 A control theory for discrete event systems, Systems and Control Group Report 8714, Department of Electrical Engineering, University of Toronto, Toronto

Multi-class queueing models for performance analysis of computer systems

KIRAN M REGE

Centre for Development of Advanced Computing, Poona University Campus, Ganeshkhind Road, Pune 411 007, India.

Abstract. Queueing models, networks of queues in particular, have been found especially useful for estimating the performance of computer systems. Networks of queues with multiple customer classes provide a flexible framework for modelling computer systems, where a rich set of analytical results and techniques are available. When because of the complexity of the system being modelled the analytical results cannot be applied directly, they often point to fairly accurate approximation schemes. In this paper, we present a brief survey of some of the important results and techniques from the theory of multi-class queueing networks. We also present a case study to illustrate how these results and techniques are used in a real-life situation where many of the modelling constraints are violated.

Keywords. Queueing networks; multi-class queues; performance analysis of computer systems.

1. Introduction

Queueing models have played a major role in performance evaluation of computer systems. Computer systems typically comprise a set of discrete resources – processors, discs, communication media etc. – for which contention takes place among competing transactions or processes. Those transactions that cannot immediately get hold of the requested resource are usually queued up in buffers until the resource becomes available. Also, transactions typically make several visits to different resources during their sojourn in the system. These characteristics make computer systems amenable to analysis using queueing models, specifically, the framework provided by networks of queues. Application of these models to computer systems has had a synergistic effect on both computer system design and queueing theory research – it has enabled system designers to use a wealth of already existing results from queueing theory to estimate system performance; on the other hand, it has also given a fillip to queueing theory research by contributing a variety of new models and opening up new research directions.

A brief history of queueing analysis of computer systems can be found in Lavenberg (1988). As stated there, the early queueing models of computer systems were single-node models with Poisson arrivals, where the CPU of the computer system was the only resource modelled. Various scheduling algorithms applicable to time-sharing systems were analysed with these models. Clearly, these models were

inadequate for analysis of more complex systems in which a large number of resources had to be explicitly modelled to account for their dependence on each other. Another limitation of these early models was that all transactions were lumped into a single homogeneous class characterized by (statistically) identical service time requirements. Queueing network models with multiple customer (transaction) classes provide a framework which overcomes many of these limitations. Of course, this larger framework too has its own limitations, and more often than not, approximations or simulations have to be resorted to in the analysis of complex computer systems.

The rest of this paper is organised as follows. In § 2, we present a brief discussion of some of the important analytical results from multi-class queueing theory. A case study from the author's experience will be presented in § 3 to illustrate how these results are used, often in conjunction with judicious approximations and simulations, in the analysis of complex computer systems.

2. Multi-class queueing models

Multi-class queueing models afford great flexibility to the system analyst attempting to build an analytical model of a computer system. The queueing network models for which analytical solutions are available (these are discussed in § 2·2) allow multiple classes of transactions with class-dependent service times and routing patterns, and even class-hopping, where a transaction is allowed to change its class and thus follow a different routing pattern on exiting a node. This enables the analyst to distinguish between different types of transactions that the computer system may be required to process. Single class models, where the analyst is forced to lump all transactions into one class with fixed routing probabilities and service-time distributions, would be too restrictive in such cases.

Sometimes, the need to use multi-class models is obvious, as when different types of transactions follow different routing patterns or have different priorities or service-time requirements. For example, the situation shown in figure 1 clearly calls for two transaction classes to represent the distinct routes followed by the two sets of transactions. The need for multiple classes may not be that obvious in the example shown in figure 2 where all transactions follow the same route S1 → S2 → S1 → out and have similar service requirements. Here, if one explicitly wants to model the fact that each transaction makes precisely two visits to S1 and one to S2, one will have to introduce two classes so that all transactions belong to class 1 as they enter S1 from their external source, change over to class 2 and move to S2 with probability

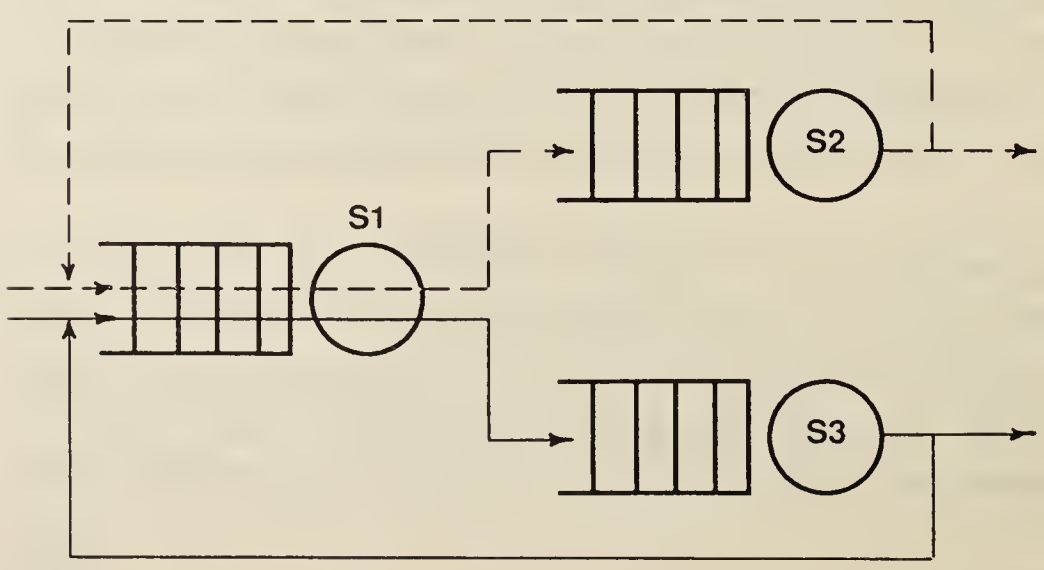


Figure 1. Two classes of transactions representing two distinct routing patterns.

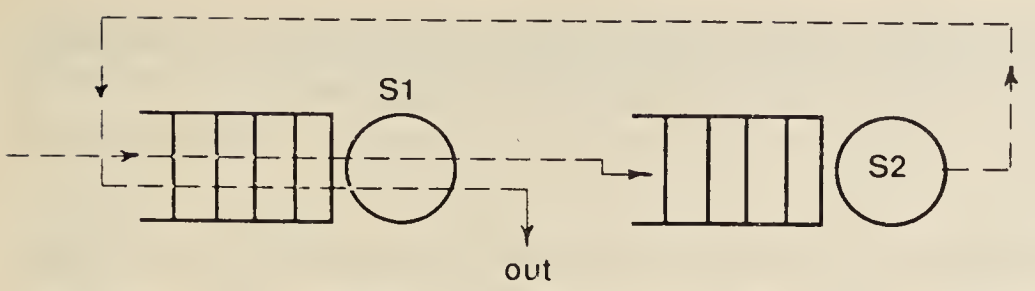


Figure 2. Single homogeneous type of transactions, each following the path S1 → S2 → S1 → out.

1 on finishing their first service at S1. When they come back to S1 after exiting S2, they belong to class 2, so that, with its different routing pattern, they exit the system with probability 1 on finishing their second service at S1. This modelling trick is required because the modelling framework allows routing to be specified in terms of fixed (class-dependent) probabilities. Therefore, if one were to lump all transactions into a single class, one would not have been able to distinguish between transactions making their first and second visits to S1 which would have resulted in many transactions making several visits to S1 before leaving the system. This discussion gives an idea of how the need for multiple classes arises in several modelling situations where such a need does not appear to exist at first glance. Conversely, it also shows how multi-class models considerably expand the scope of the queueing network formalisms.

2.1 *Single-node models with multiple classes*

Most of the single-node queueing models used for computer system analysis fall in the M/G/1 paradigm (Kleinrock 1975). This is but natural as the analysts usually feel more comfortable with the Poisson assumption for the arrival process with the flexibility afforded by the M/G/1 models to choose a general service-time distribution. The G/M/1 paradigm (and its generalisations such as the SM/M/1 model) although well researched, appears to be much less popular as it imposes the restriction that the service times be exponentially distributed.

The M/G/1 'first-come-first-served' (FCFS) queue has been analysed in classical queueing theory. The main results there are the steady state probability mass function of the number of customers (transactions) in the system or the probability distribution function of the waiting time of the customer, both of which are obtained in the form of their Laplace transforms (or Z-transforms). These are given by the famous Pollaczek-Khinchin equations. The M/G/1 queue with 'last-come-first-served-pre-emptive-resume' (LCFS-PR) discipline has also been part of classical queueing theory literature and lends itself to elegant analysis involving the concept of "busy periods" (Kleinrock 1975).

If the arrival process consists of transactions belonging to several classes, each with its own service requirements, the M/G/1 model still holds if the component arrival processes are independent Poisson and if the server makes no discrimination in providing service to different classes of transactions. This is because the aggregate process again turns out to be Poisson and the service-time distribution, a convex combination of the individual service-time distributions.

An important model within the M/G/1 paradigm is the one where the server provides service to the transactions waiting there in a round-robin (RR) manner with a finite service quantum. A numerical procedure for analysing this model has been given in Wolff (1970). A mathematical idealisation of this model where the service quantum shrinks to zero has been called the processor-sharing (PS) model. This lends

itself to elegant analysis as in Kleinrock (1976), Ott (1984) and Yashkov (1983), for example. Again, analysis of the M/G/1 model with the PS discipline remains valid with multiple transaction classes as long as the server makes no discrimination on the basis of transaction class.

If one were to give different priorities to transactions belonging to different classes, the waiting time distribution would obviously be class-dependent. The standard single-class M/G/1 analysis would not easily extend to this case even though the aggregate arrival process is still Poisson. However, the M/G/1 model with both preemptive-resume (PR) and non-preemptive (NP) disciplines can be analysed to obtain closed form solutions for the waiting-time distributions of transactions belonging to different classes. The classical analysis employing the concept of busy cycles can be found in Conway *et al* (1967). The level-crossing approach of Brill and Posner (1977) can also be used to obtain these expressions.

2.2 *Networks of queues with multiple classes*

Single-node models are useful when the system has one important resource which needs to be modelled carefully. These models are totally inadequate for analysing complex systems characterized by multiple sets of resources in which transactions make several trips between these resources during their sojourn through the system. The framework of queueing networks meets many of the modelling requirements of these systems. Where this framework cannot provide an exact solution, it points to certain approximations which lead to fairly accurate estimates of certain performance measures.

The first important result for networks of queues was due to Jackson (1963), who showed that single-class open networks of queues consisting of FCFS servers with exponential service-time distribution have a product form solution for their equilibrium state probabilities. (Open networks of queues are those where the transactions are generated by an external source and spend a finite period of time in the system before departing. Closed networks are those where the transactions permanently remain within the network, circulating from node to node. Closed networks are particularly useful in the analysis of multi-programmed time-sharing systems.) Gorden & Newell (1967) showed that single-class closed networks of queues with exponential servers also have a product form solution except for a normalising constant. Both Jackson and Gordon–Newell networks are special cases of the so-called BCMP networks which will be discussed in some detail below.

Probably the most important analytical result in the theory of queueing networks is due to Baskett *et al* (1975) who combined many of the previously known results and brought them together in a single framework of 'local balance' networks. These networks (called BCMP after the names of the authors), include open, closed and mixed systems consisting of four types of nodes – FCFS, LCFS-PR, PS and infinite-server (IS) nodes. Multiple transaction classes are allowed, routing is class-dependent and probabilistic; also, transactions are allowed to change classes in a probabilistic manner. The FCFS nodes in these networks have the restriction that all transaction classes have to have the same service-time distribution there which is required to be exponential. At all other nodes different classes are allowed to have different general service-time distributions. All classes have the same priority at all nodes.

It will be necessary to introduce some notation to state the main result of Baskett *et al* (1975). So, let M denote the number of nodes (queues) in the network, K the

number of transaction classes and n_{mk} the number of transactions of class k at node m for $1 \leqslant m \leqslant M$ and $1 \leqslant k \leqslant K$. Let N denote the state of the network where $N = (n_1, n_2, \ldots, n_M)$, for $1 \leqslant m \leqslant M, n_m = (n_{m1}, n_{m2}, \ldots, n_{mK})$. Then, the main result of Baskett *et al* (1975) states that under certain reachability conditions for the FCFS nodes, the equilibrium probability $P(N)$ of the network being in state N is given by

$$P(N) = P_1(n_1) \cdot P_2(n_2) \ldots . P_M(n_M)/G,$$

where, for $1 \leqslant m \leqslant M, P_m(n_m)$ is calculated by analysing the mth queue in isolation with the transactions belonging to the various classes present at that queue being fed by independent Poisson processes with arrival rates proportional to the corresponding visit ratios. The visit ratios are computed by solving a set of linear equations involving routing and class-hopping probabilities.

The product form nature of the solution of BCMP networks enables the analyst to calculate equilibrium state probabilities without having to solve the corresponding Markov chain. This is a great advantage as even for moderately large networks the state-space of the Markov chain becomes extremely large making a direct computational solution impossible. It is easy to see that many performance measures of interest, e.g., throughput, mean queue lengths, mean waiting times etc. can be directly calculated from the equilibrium state probabilities. For complete distributions of sojourn times, however, just a knowledge of state probabilities is inadequate.

The expression for the equilibrium state probabilities cannot be complete unless the remaining constant G is calculated. To calculate G the product $P_1(n_1) \ldots P_M(n_M)$ needs to be summed over all possible network states N. For large networks with many classes and nodes, this computation can be extremely time-consuming. Buzen (1973) has developed an efficient algorithm (called the convolution algorithm) to calculate G for single-class closed networks. It has been generalized later to multi-class closed networks by Reiser & Kobayashi (1975).

Another important result in the multi-class queueing network literature is the mean-value analysis (MVA) algorithm presented in Reiser & Lavenberg (1980). In this algorithm, instead of determining the equilibrium state probabilities from which performance measures such as mean waiting times and queue length and throughput can be calculated, the latter are computed directly in a recursive manner.

The so-called Norton's theorem for product form queueing networks (Chandy *et al* 1975) also represents an important and useful result in queueing network literature. This theorem states that if a product form network is divided into two subnets, then for the analysis of one of the subnets the other may be replaced by a single node with a flow-equivalent server having a state dependent service rate. This result is often used to obtain approximate solutions of almost-product-form networks. The case study presented in § 3 also uses an approximation scheme involving this result.

As stated earlier, the routing probabilities in a BCMP network can depend on the transaction class, the node from which the transaction is about to depart, and the destination node. In models of some complex systems the routing probabilities can depend on other variables besides these three. The BCMP framework, obviously, breaks down in these cases. In Towsley (1980) one finds a useful extension (for single-class systems) where a restricted form of state-dependent routing is allowed in certain subnets consisting of parallel branches with common entry and exit points. It has been shown there that the resultant equilibrium state probabilities in these networks have a modified product form solution. An interesting and rather unexpected

application of this result can be found in Rege (1988) where it is used in the analysis of a blocking scheme for a manufacturing line.

Although BCMP networks with multiple transaction classes provide a flexible and fairly general framework for analysis of computer systems, they have their obvious limitations such as the restrictive nature of the queueing disciplines allowed, lack of priorities etc. Thus, for many real-life systems, an approximate analysis or simulation is the only recourse. Performance analysis literature is replete with many clever approximation schemes built around analytical results such as those presented earlier. It is impossible to even enumerate these efforts, many of which have limited applicability. The case study presented in the next section will illustrate the 'practice' or 'art' of performance analysis which involves choosing clever combinations of analytical results, approximations and discrete-event simulations.

3. A case study

As stated at the end of the previous section, most of the large real-life systems do not fit into the framework of BCMP networks for which analytical results are available. An attempt to "force-fit" them into this framework by ignoring or glossing over some system features would result in over-simplification, yielding unreliable performance estimates. If one were to incorporate into the model most of the important system features, the complexity of the model would be such that a detailed simulation of the system would be needed for estimation of system performance. While a simulation affords a great deal of flexibility to the analyst, it usually requires a large amount of processing time to provide good estimates of the desired performance measures. Thus, the analyst is required to devise several tricks to reduce the simulation time. The following example from the author's experience will illustrate this point. This case study was reported in Becker *et al* (1984).

The system analysed in Becker *et al* (1984) was called FACS which was a distributed computer system consisting of several component systems – SOAC, LFACS, COSMOS, LAC and WM as shown in figure 3. SOAC and LFACS resided on the same mainframe

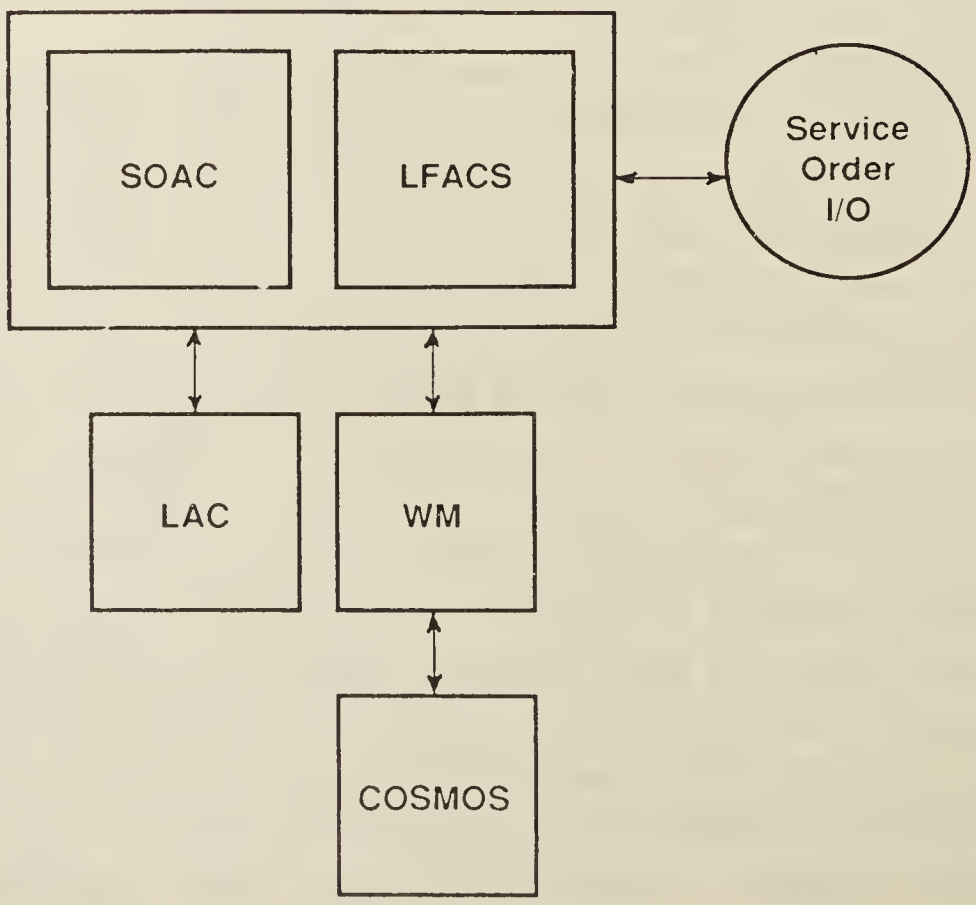


Figure 3. FACS architecture.

computer which consisted of multiple central processing units and disks. This mainframe was the principal object of the analysis.

The input to this system came in the form of "service orders" which had a distinct time-varying arrival pattern over a working day. The service-orders fell into different classes and had different due date requirements. Each service-order decomposed into one of several possible sequences of functional activities, each of which was further divided into a sequence of processing units called transactions. The transactions were grouped into five categories for measurements and had three priority levels – the highest of these was called "immediate", the rest were called "deferred". The mainframe had a limited multiprogramming capability. A priority based admission scheme was used to allow transactions access to the CPU and disk subsystem. The process scheduler within the CPU system also gave a preemptive priority to the immediate transactions over the deferred. The overall objective was to provide a tool that could be used to size the system to maximize its utilization without violating certain performance constraints.

Clearly, many features of the system, such as the time-varying arrival pattern, could not be included in the model without resorting to a simulation. The multi-class BCMP framework is simply incapable of handling this requirement. If one were to build a detailed simulation of this sytem which would simulate even the trips made by transactions between the CPU and the disks, the processing time requirement would be inordinately large as each transaction typically made 50 to 100 such trips before leaving the system, thus giving rise to a large number of "events". It was, therefore, decided to follow a middle path, that is, a hybrid approach was chosen. Thus, an analytical model was built for the CPU and disk subsystem, which was

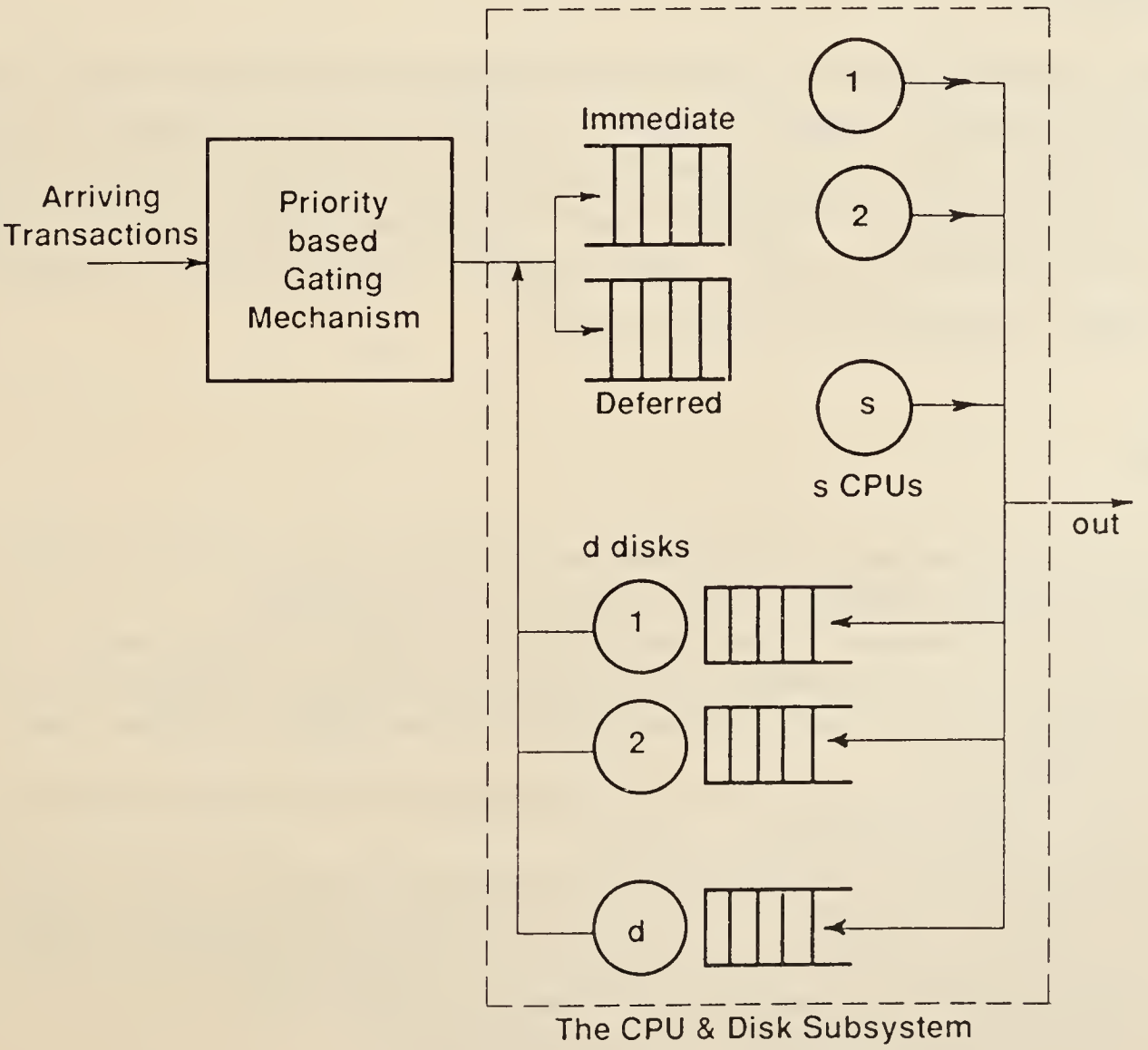


Figure 4. A queueing network model of the mainframe computer.

analysed off-line to obtain certain "service-time expansion factors". These were then used in a simulation of the rest of the system to calculate the effect of the interactions between different transactions on their response time. Thus, the lowest-level events in the simulation were the entry and exit of transactions from the computer system rather than their visits between the CPU and disks. This reduced the total number of events in the simulation by a factor of about a hundred.

Figure 4 shows a queueing network representation of the CPU and disk subsystem which was to be solved analytically. Because of the priority given at the CPU to the immediate transactions over the deferred, BCMP results could not be used to derive the equilibrium probability distribution. Therefore, an approximation had to be used. Taking a cue from Norton's theorem for product form networks, the disk subsystem was replaced with a state-dependent server whose rates were determined for each possible state. Given this state-dependent server, the overall network could be represented by a simple 2-dimensional Markov process with a finite state-space. This Markov process had a structure which made calculation of its steady-state probabilities rather easy. The service time expansion factors could then be calculated in terms of these probabilities. Since these factors were required for each possible combination of immediate and deferred transactions inside the CPU and disk subsystem, the above procedure had to be repeated that many times to obtain a complete set of these factors.

The above approach worked well, both in terms of the accuracy of its results and run-time requirements – when the model was validated with some measured data, most of the performance estimates were within 10% of their measured values. At the same time, the hybrid approach kept run-time requirements within manageable limits.

4. Conclusion

Multi-class queueing networks present an elegant and useful framework for analysis of computer systems. A wealth of results is available for this framework. However, conditions under which exact analytical results are available within this framework are often too restrictive for modelling today's complex computer systems. A judicious combination of analytical results, approximations and simulation is needed to deal with the many complexities of these systems.

References

Baskett F, Chandy K M, Muntz R R, Palacios-Gomez F 1975 Open, closed and mixed networks of queues with different classes of customers. *J. Assoc. Comput. Mach.* 22: 248–260

Becker S T, Rege K M, Sengupta B 1984 A modelling methodology for sizing a computer based system in a netted environment. *Proceedings Assoc. Comput. Mach. Sigmetrics Conference*

Brill P H, Posner M J M 1977 Level crossings in point processes applied to queues: Single server case. *Oper. Res.* 25: 662–674

Buzen J P 1973 Computational algorithms for closed queueing networks with exponential servers. *Commun. Assoc. Comput. Mach.* 16: 527–531

Chandy K M, Herzag U, Woo L 1975 Parametric analysis of queueing networks. *IBM J. Res. Dev.* 19: 36–42

Conway R W, Maxwell W L, Miller L W 1967 *Theory of scheduling* (Reading, Mass: Addison-Wesley)

Gordon W J, Newell G F 1967 Closed queueing systems with exponential servers. *Oper. Res.* 15: 254–265

Jackson J R 1963 Jobshop-like queueing systems. *Manage. Sci.* 10: 131–142

Kleinrock L 1975 *Queueing systems* (New York: John Wiley) Vol. 1

Kleinrock L 1976 *Queueing systems* (New York: John Wiley) Vol. 2

Lavenberg S S 1988 A perspective on queueing models of computer performance, in *Queueing theory and its applications*–Liber Amicorum for J W Cohen: CWI Monograph 7 (Amsterdam: North-Holland)

Ott T J 1984 The sojourn time distribution in the M/G/1 queue with processor sharing. *J. Appl. Probab.* 21: 360–378

Rege K M 1988 Approximate analysis of serial manufacturing lines with buffer control. *Inf. Decision Technol.* 14: 31–43

Reiser M, Kobayashi H 1975 Queueing networks with multiple closed chains: Theory and computational algorithms. *IBM J. Res. Dev.* 19: 283–294

Reiser M, Lavenberg S S 1980 Mean-value analysis of closed multi-class queueing networks. *J. Assoc. Comput. Mach.* 27: 313–322

Towsley D F 1980 Queueing models with state-dependent routing. *J. Assoc. Comput. Mach.* 27: 323–337

Wolff R M 1970 *SIAM J. Appl. Math.* 19: 566–574

Yashkov S F 1983 A derivation of response time distribution for the M/G/1 processor sharing queue. *Prob. Control Inf. Theory* 12: 133–148

Stability of multiple access communication networks

VINOD SHARMA

Department of Electrical Engineering, Indian Institute of Science, Bangalore 560 012, India

Abstract. The problem of stability of different multiple access schemes in computer networks is discussed. The results for the main multiple access schemes – slotted ALOHA, controlled ALOHA, CSMA, CSMA/CD, polling systems and collision resolution algorithms are surveyed. The techniques used for the stability of these systems have been emphasized.

Keywords. Multiple access schemes; communication networks; channel access; stability.

1. Introduction

In data communications several resources usually share a common channel of communication. Traditionally, in telephone or radio networks time division multiple access (TDMA) or frequency division multiple access (FDMA) has been used to access the channel. These methods of channel access are efficient if the number of users is not very large and/or the communication traffic is not very bursty. But with the advent of large-scale use of computers it was felt that dedicating a time slot or a frequency band to each user is not very efficient because in such environments both these assumptions are generally violated. This could lead to unusually large delay in the communication and the channel may remain idle for a large fraction of time. This led researchers to look for new methods to share a channel.

The earliest new scheme for sharing channels was ALOHA. An improvement over it was slotted ALOHA. Since then a large number of techniques have been developed. Good reviews are available in Sachs (1988), Bertsekas & Gallagher (1987). The purpose of this paper is to explain the main methods of multiple access that have been developed over the years and, more importantly, to present the most useful techniques that have been used to prove the stability of the channel under these methods.

We shall model the multiple access system as a stochastic system where the input traffic generated at the users for transmission is viewed as a stochastic process. In the literature the term stability has been used to imply different things. For us, the stability would usually mean the existence of stationary distributions of some system parameters of interest (e.g. queue lengths, delays) and the convergence to stationary distributions starting from any initial distribution. But we shall also briefly present other forms of stability like existence and convergence of moments of system parameters, structural stability, boundedness of sample paths of system parameters etc.

The paper is planned as follows. Section 2 describes the general multiple access

system and the assumptions made. A unified stability analysis is given in § 3 for the slotted ALOHA and TDMA systems. Section 4 discusses the controlled ALOHA system. First the instability of the infinite user model is proved and then various schemes to stabilize it have been mentioned. Again, in § 5 for CSMA and CSMA/CD the instability of the infinite user model is shown and then some work concerning its stabilization has been described. Section 6 discusses the polling systems and § 7 the collision resolution schemes. Section 8 briefly mentions some hybrid schemes which combine the random access and reservation schemes. Methods used to prove the stability of the various systems have been highlighted, the ergodicity of appropriate Markov chains being the predominant theme.

2. Multiple access system

There are $1 < M \leqslant \infty$ users which share a common channel to communicate with each other. The messages are assumed packetized and usually having fixed packet lengths. The time axis will at times be slotted, i.e., it will be divided into slots of fixed size sufficient to send a packet if the packets are of the same lengths. In the slotted case the transmission of a packet can start only at the beginning of a slot. A user sends at most one packet at a time if it has something to transmit. The users will be assumed to have finite or infinite buffers to store the packets until successful transmission. When $M = \infty$, each user will be assumed to have a single buffer. If the transmission times of two or more packets overlap then the transmission is unsuccessful and the transmitters will send those packets again. On successful transmission the packets are removed from the buffers. It is assumed that the users can detect a collision or a successful transmission. The exact details of different multiple access methods will be presented at the appropriate places.

The stability of multiple access systems is generally studied either with $M < \infty$ and each user having an infinite buffer or with $M = \infty$. Both of these systems are idealization of the real systems where $M < \infty$ and the buffers at each user are finite. The relations between the stability for $M < \infty$ and $M = \infty$ assumptions have been studied by Paterakis *et al* (1987).

3. Slotted ALOHA and TDMA

In this section we present a technique by Loynes (1962) which has been used to demonstrate the stability of many systems. The advantage of this method is that it is quite intuitive and gives the existence of and convergence to stationary distributions under general stationary assumptions. We first explain the method in general and then apply it to slotted ALOHA and TDMA systems.

Let $\{x_k\}$ be a sequence of stationary random vectors with values in R^n and let $\{z_n\}$ be a sequence with values in R^m defined by (we call $\{x_n\}$ the input sequence and $\{z_n\}$ the state sequence)

$$z_{n+1} = f(z_n, x_n), \tag{1}$$

where f is a measurable function. The following is Loyne's lemma.

Lemma 1. *If $f(z, x)$ is a nonnegative monotonically nondecreasing function in the first variable z for each x and is also continuous in z then* (1) *has a stationary solution.*

Proof. Let us denote by z_k^{-n} the state value at time k if the system started at time $-n$ with initial state 0. Then it is easy to see that

$$z_k^{-n-2} \geqslant z_k^{-n-1} \geqslant z_k^{-n} \quad \text{a.s.}$$

Hence as $n \to \infty$, z_k^{-n} converges almost surely (a.s.) to a limit z_k. It can easily be shown that the $\{z_k\}$ sequence so obtained is stationary and satisfies (1) a.s.

Remark 1. We can also show that if the input sequence $\{x_m\}$ is ergodic then so is the $\{z_n\}$ sequence obtained in the above lemma. Of course, it is possible that the limit sequence $z_k = \infty$ a.s. Therefore, to obtain an a.s. finite sequence we need to have extra conditions which we mention when we consider the particular systems.

Remark 2. It is easy to see that the distribution of z_k^{-n} is the same as that of z_{n+k}^0. Therefore, since $z_k^{-n} \xrightarrow{W} z_k$ in the above lemma, we also have $z_{n+k}^0 \xrightarrow{W} z_k$ as $n \to \infty$. Thus if the system starts empty at time 0, then it converges in distribution to z_k. This can be shown for all finite dimensional distributions also.

Remark 3. The above lemma does not guarantee the uniqueness of the stationary distributions. But it can be shown that if more than one stationary distribution exists then the sequence obtained in lemma 1 is stochastically the smallest stationary distribution of (1).

Remark 4. The condition of monotonicity in the above lemma is quite strong and is not satisfied in many computer network systems.

Many practical systems can be modelled more appropriately to have periodic inputs, i.e., the finite dimensional distributions of $\{x_n, x_{n+1}, \ldots, x_{n+m}\}$ and $\{x_{n+d}, x_{n+1+d}, \ldots, x_{n+m+d}\}$ are the same for some integer $d \geqslant 1$ and for all n and m. For this periodicity, we show now that the state equation $\{z_n\}$ in (1) has periodic stationary distributions with period d under the assumptions of lemma 1.

Under the assumption of periodicity,

$$z_k^{-n} \overset{\text{dist}}{=} z_{k+d}^{-n+d},$$

where $X \overset{\text{dist}}{=} Y$ indicates that the random variables X and Y have the same distribution. Since $z_k^{-n} \xrightarrow{W} z_k$ and $z_{k+d}^{-n+d} \xrightarrow{W} z_{k+d}$, z_k and z_{k+d} have the same distribution. The same can be said for all finite dimensional distributions. Hence, we can also show that if the system starts empty at time $n = 0$, it converges to these periodic stationary distributions. Similar ideas have been expressed in Rolski (1981) and Bambos & Walrand (1989) also.

Now we apply the above results on a multiple access system which has been analysed in Prathima (1991). This is a new scheme for multiple access that combines the advantages of slotted ALOHA and TDMA. The results for stationary and periodic stationary cases for the slotted ALOHA and TDMA will emerge as special cases.

We assume there are $1 < M < \infty$ users each having infinite buffer to store the packets. The packets are of fixed size and the time axis is slotted. The M users are

divided into L groups $G_1, G_2, \ldots G_L$ (they don't have to be disjoint). Each slot is assigned to one of the groups in a periodic fashion. We take the period as d slots. The question of how to form the groups, how large d should be and how many slots should be assigned to a group in a period is an optimization problem which depends upon the traffic rates of different users. This problem is addressed in Prathima (1991) but presently we assume that the assignment has been made. In the slot assigned to a group, all the users which have packets to transmit at the beginning of that slot transmit a packet with a certain probability. If more than one packet gets sent then there is a collision and those packets will be tried again in the next assigned slot.

We use the following notation:

kth slot = time interval $(k, k+1)$;
$x_k(i)$ = number of new packets generated in the $(k-1)$ slot at the user i;
$t_k(i)$ = 1, if ith user would transmit a packet in the kth slot if it has a packet at time k
= 0, otherwise;
$z_k(i)$ = queue length at the ith user at time k without including $x_k(i)$;
1_A = indicator function of set A;
$y_k(i)$ = 1, if the ith user has a successful transmission in the kth slot
= 0, otherwise.

Then the system equation can be written as

$$z_{k+1} = z_k + (x_k - y_k)$$

where $z_k = (z_k(1), z_k(2), \ldots, z_k(M))$ and similarly for x_k and y_k. We can write this equation as

$$z_{k+1} = f(z_k, x_k, t_k).$$

The random variables t_k take care of the assignment of the slots to different groups.

We can see that the function f is monotonically nondecreasing in z and is nonnegative. Thus, if $\{(x_k, t_k)\}$ is a periodic stationary process, then there exists a periodic stationary distribution of $\{z_k\}$ and starting at time $t = 0$ with $z_k = 0$, the finite dimensional distributions of $\{z_k\}$ converge to the periodic stationary distributions. For the finiteness of the stationary distributions (proper random variables) we need extra conditions on the rate of the input traffic. It can be shown, as in Sharma (1988), that if $Ex_k < \infty, \Sigma_{k=1}^{d} E(x_k - y_k) < 0$ and the input traffic $\{(x_k, t_k)\}$ is ergodic, the periodic stationary distributions of $\{z_k\}$ are proper. Also, as in Sharma (1988), it can be shown that if the system starts at $t = 0$ with nonzero finite a.s. initial state then the random variables $\{z_k\}$ remain bounded a.s. Other remarks in Sharma (1988) remain relevant for this system as well.

It is easy to see that if $\{t_k\}$ is stationary then the above system becomes slotted ALOHA with periodic input, if $\{(x_k, t_k)\}$ is stationary then it is slotted ALOHA with stationary input, if $\Sigma_{i=1}^{M} t_k(i) = 1$ for all k and t_k is deterministic then it is TDMA with periodic input.

Various other results have been obtained on the slotted ALOHA and TDMA systems. Slotted ALOHA with a finite number of users and infinite buffers has been studied by Saadawi & Ephremides (1981) by forming two coupled Markov chains. Stability using Markov chains has also been shown by Tsybakov & Mikhailov (1979). Both these papers assume the input traffic $\{x_n\}$ to be i.i.d. Within this framework most of the ergodicity and nonergodicity results have been summarized in Szpankowski (1986).

For nonhomogeneous, nonperiodic input traffic, analysis has been provided in Sharma (1987). A comparative analysis of TDMA and FDMA is provided in Rubin (1979). The structural stability of slotted ALOHA has been studied in Nelson (1984) and Onozato & Noguchi (1985) via catastrophe theory. That the slotted ALOHA system with $M = \infty$ and unbuffered users is unstable for all input traffic rates has been proved by Rosenkrantz & Towsley (1983) by martingale techniques. We present this result in detail in the next section.

4. Controlled ALOHA

In the last section we studied the slotted ALOHA system in which the retransmission probabilities remain constant. As mentioned there, it has been shown in Rosenkrantz & Towsley (1983 and the earlier references therein) that for an infinite user model the slotted ALOHA system is unstable for any positive total input traffic, i.e., it has been shown that the total number of blocked users goes to infinity with probability 1. This result, which was first noticed in the mid-seventies, has evoked a lot of research to control the retransmission probabilities of the users so that the system becomes stable for some nonzero input traffic (see Hajek 1985, Merakos & Kazakos 1985, Bertsekas & Gallagher 1987). The methods used for the study of such systems are quite different from those in the last section. In Merakos & Kazakos (1985) and Hajek (1985) stochastic approximation methods have been used. Theory of Markov chains and large deviations has been used in Lim & Meerkov (1987). Large deviation results have been used in Cottrell *et al* (1983) also. Martingale techniques have been used in Goodman *et al* (1988) to prove the stability of exponential backoff schemes. In § 4.1 we show the above mentioned instability of slotted ALOHA for $M = \infty$. Section 4.2 will present a control scheme to stabilize the slotted ALOHA under these circumstances.

4.1 *Instability of infinite user slotted ALOHA*

We assume that the number of users $M = \infty$, each having a single buffer. Thus if a user has a packet stored then it cannot generate any more packets. Let $\{x_k\}$ be a sequence of i.i.d. random variables where x_k is the number of new packets generated in the system in the $(k-1)$th slot. Let z_k represent the number of blocked users at time k, i.e., number of users which have packets that have been transmitted unsuccessfully earlier. A new packet generated in slot $k-1$ is transmitted in slot k with probability 1. A blocked user transmits its packet in the next slot with probability p. Then the $\{z_k\}$ sequence is a Markov chain with transition probabilities P_{ij} given as below [in the following $P(x_k = i) = \lambda_i$],

$$
\begin{aligned}
P_{00} &= \lambda_0 + \lambda_1; \quad P_{01} = 0; \quad P_{0j} = \lambda_j; \\
P_{i,i-1} &= \lambda_0 ip(1-p)^{i-1}, \quad i \geqslant 1, \\
P_{ii} &= \lambda_1(1-p)^i + \lambda_0(1 - ip(1-p)^{i-1}), \quad i \geqslant 1, \\
P_{i,i+1} &= \lambda_1(1-(1-p)^i), \quad i \geqslant 1, \\
P_{ij} &= \lambda_{j-1}, \quad j \geqslant i+2, \quad i \geqslant 1.
\end{aligned}
$$

The following theorem has been used in Rosenkrantz & Towsley (1983) to show the transience of Markov chain $\{z_k\}$.

Theorem 1. *If an irreducible, aperiodic Markov chain* $\{z_k\}$ *admits a nonconstant, nonnegative, bounded function* f *on its state space* $\{0, 1, 2, \ldots\}$ *such that*

$$E[f(z_{k+1})|z_k] \leqslant f(z_k),$$

then the Markov chain is transient.

When $0 < \lambda_0 < 1$ then $\{z_k\}$ is aperiodic. We further assume that $\lambda_0 + \lambda_1 < 1$, otherwise the chain will be ergodic. Under these conditions a function f has been constructed in Rosenkrantz & Towsley (1983) which satisfies the conditions of theorem 1 and hence $\{z_k\}$ is transient.

4.2 *Controlled* ALOHA

We have seen in §4.1 that for the infinite user case the slotted ALOHA system is unstable if the retransmission probability p is kept fixed. This happens because if at any time the number of blocked users increases, since p is fixed, the probability that only one user transmits is very low. Therefore, as new packets are generated, the number of blocked packets keeps increasing with a high probability. Hence, attempts have been made to adaptively change p as a function of the number of blocked users z_k so that the probability of successful transmission remains high (Merakos & Kazakos 1985; Hajek 1985; Lim & Meerkov 1987; Tsitsklis 1987).

Now we present a scheme, given in Tsitsklis (1987) to stabilize the slotted ALOHA. As in §4.1, x_k is the number of new packets generated in slot $k-1$ and z_k is the number of blocked users at time k (beginning of slot k). Let y_k be the number of attempted transmissions in slot k. At the beginning of each slot every station has an estimate $\hat{z}_k$ of the backlog, which it updates in each slot as below (the users get the feedback from the channel whether a slot was empty, there was successful transmission or a collision).

$$\hat{z}_{k+1} = \max\{1 + \hat{z}_k - 1 + \hat{\lambda}\}, \quad \text{if } Y_k < 2,$$
$$\hat{z}_{k+1} = \hat{z}_k + [1/(e-2)] + \hat{\lambda}, \quad \text{if } Y_k \geqslant 2,$$

where $\hat{\lambda}$ is the estimate of λ at each user, λ being the expected value of x_k. Actually, $\hat{\lambda}$ can be replaced by e^{-1} and we can still obtain the stability and maximum throughput. The motivation for the updates of z_k is provided in Bertsekas & Gallagher (1987) and Rivest (1987).

We define $S_k = (z_k, \hat{z}_k)$. Then under the condition of $\{x_k\}$ being i.i.d., S_k is a Markov chain. Now using the techniques of Hajek (1982) a Lyapunov function can be constructed to prove the following theorem.

Theorem 2. *If* $0 < \hat{\lambda} \leqslant e^{-1}, 0 < \lambda < e^{-1}$ *and* $\lambda \leqslant \hat{\lambda}$, *then the Markov process* S_k *defined above is geometrically ergodic (Tsitsklis 1987).*

In the schemes mentioned in Merakos & Kazakos (1985), Hajek (1985), Tsitsklis (1987) and Rivest (1987) each station has to continuously monitor the channel and update the backlog estimates etc. to follow the algorithm and their estimates should agree with each other for stability. To overcome this disadvantage, other algorithms have been proposed to control the slotted ALOHA (Aldous 1987; Hastad *et al* 1987; Goodman *et al* 1988). In Aldous (1987) it has been shown that for $M = \infty$ the Markov chain of backlogs $\{z_n\}$, when the retransmission probabilities p of the users are

exponentially decreased, is transient. In Goodman *et al* (1988), for number of users $M < \infty$, each with infinite buffer, under the exponential backoff, the Markov chain of queue lengths at each station has been shown to be positive recurrent if the traffic rate is below certain limits. These results have been generalized in Hastad *et al* (1987). Of course for finite user models the uncontrolled slotted ALOHA itself is stable. But the results of Hastad *et al* (1987) show that by controlling the retransmission probabilities we can increase the maximum stable throughput of the system.

The possibility of successful transmission of more than one packet in a slot has been considered in several papers (this is the "capture" effect). For such a slotted ALOHA network, control algorithms have been analysed in Ghez *et al* (1989).

The delay analysis of controlled ALOHA has been considered in Thomopoulos & Sunder (1987) and Clare (1986).

5. CSMA and CSMA/CD

The multiple access scheme CSMA was first proposed and analyzed by Kleinrock & Tobagi (1975). Its stability and control was discussed in Tobagi & Kleinrock (1977). Since then a large number of papers have been devoted to this subject (Bertsekas & Gallagher 1987). This scheme has proved very efficient when the propagation delays between different users are much less than the transmission times of the packets. An improvement over CSMA is CSMA/CD (Bertsekas & Gallagher 1987; Tobagi & Hunt 1987; Sachs 1988).

In CSMA, $M \leqslant \infty$ users share a common channel where each user can sense whether the channel is idle or busy. Whenever a user has a packet to send it first senses the channel for sometime (the time depends upon the propagation delays). If channel is free during this time then it sends its packet with probability $0 \leqslant p \leqslant 1$. If the channel is busy then it reschedules its sensing of the channel. When it next senses the channel depends upon the protocol assumed. Even after sensing that the channel is free, there is a possibility of a collision. In CSMA the user sends the complete packet even if there is a collision. But in CSMA/CD (CD stands for collision detection) after it senses a collision, the user aborts the transmission and hence frees the channel earlier than in CSMA.

5.1 *Instability of CSMA and CSMA/CD*

In this section we prove that the infinite user CSMA and CSMA/CD channels are unstable. The analysis follows Medith & Lea (1983) where the Markov chain of blocked users is formed and is shown to be transient by using theorem 1.

There are $M = \infty$ users each with a single buffer. The time axis is partitioned into minislots of lengths which equal the one-way propagation delay of the whole channel. The data packets are of fixed length and require a transmission time of $T > 1$ minislots. In nonpersistent CSMA, if a user has a packet ready for transmission, it senses the channel. If the channel is sensed idle, it transmits the packet in the next slot. If it is found busy, the user schedules its transmission for some future minislot according to a probability distribution. In that slot if it again senses the channel busy it repeats the procedure. A blocked user senses the current slot with probability $0 < f \leqslant 1$ and thereupon proceeds as above. The nonblocked users are assumed to generate new packets with Poisson distribution and mean rate λ packets/minislot.

The actual time taken in transmission for CSMA is $1 + T$ minislots whether the transmission is a success or a failure (collision). Between two successive transmissions there is at least one idle slot. We consider the backlogged process at the instants of the end of the first idle minislot after a transmission and denote by z_i the number of blocked users at these instants. This is a Markov chain. The transition probability of this Markov chain can be easily calculated under the above assumptions. Then, using theorem 1, the following result can be obtained.

Theorem 3. *The Markov chain $\{z_n\}$ of block users at the embedded points is transient (Medith & Lee 1986).*

A similar result for CSMA/CD has also been obtained in Medith & Lea (1983). As for slotted ALOHA, this instability result has led researchers to obtain control schemes for CSMA and CSMA/CD also. We consider some of these results in the next section.

The delay analysis for CSMA and CSMA/CD has been studied in Beuerman & Coyle (1988) and Tobagi (1982). Matrix geometric methods have been used in Beuerman & Coyle (1988), while Tobagi (1982) uses Z-transform techniques.

5.2 *Control of CSMA and CSMA/CD*

To stabilise the CSMA and CSMA/CD adaptive algorithms changing the retransmission probabilities etc. have been proposed (Tobagi & Kleinrock 1977; Medith & Lea 1983; Rubin 1983; Coyle & Liu 1985) just as for slotted ALOHA. Medith & Lea (1983) consider only the slotted system, while Coyle & Liu (1985) consider only the unslotted (continuous time) system and both cases are considered by Rubin (1983).

Medith & Lea (1983) control CSMA and CSMA/CD by varying the retransmission probabilities f of the blocked users as $f = \alpha/n$ where n is the backlog. Of course n will not be known to the users and hence will have to be estimated. The estimation schemes which have been used for controlling slotted ALOHA can be useful in this case also (see details later). Under the adaptive scheme of $f = \alpha/n$ Medith & Lea (1983) have shown the ergodicity of the Markov chain $\{z_n\}$ formed in § 5.1. For this they use the following Pake's lemma (Pake 1969).

Lemma 2. Let $\{x_i\}$ be an irreducible, aperiodic Markov chain whose state space is the set of nonnegative integers. Then, the following conditions are sufficient for the chain to be ergodic:

$$|E[x_{i+1} - x_i | x_i = n]| < \infty,$$

and

$$\limsup_{n \to \infty} E[x_{i+1} - x_i | x_i = n] < 0.$$

Using Lemma 2 it has been shown (Medith & Lea 1983) that if

$$\lambda T[1 + (T+1)(1 - e^{-(\alpha+\lambda)})] < (\alpha + \lambda)\, T e^{-(\alpha+\lambda)},$$

then Markov chain $\{z_n\}$ for CSMA is ergodic. A similar expression has been obtained for CSMA/CD also. From this condition, α can be appropriately chosen so as to obtain ergodicity for different arrival rates λ.

Coyle & Liu (1985) have chosen the same adaptive scheme $f = \alpha/n$ and provide conditions for ergodicity, null-recurrence and transience of the Markov chain for the

unslotted CSMA/CD. The system has been studied for exponential as well as phase type distributions and explicit closed form expressions obtained using the matrix-geometric techniques of Neuts (1981).

More recently, control schemes have been proposed which do not require the stations to know the total system backlog (Thomopoulos 1988; Cunningham & Medith 1988). In (Thomopoulos 1988) minimum mean square error predictor has been used for the number of blocked users z_n as below (the updating is done at the end of the idle period, the successful transmission period and the collision period).

$$\begin{aligned}\hat{Z}_{n+1} &= \hat{Z}_n - \alpha/(\alpha + \lambda_{\text{eff}}), \quad \text{if successful transmission,}\\ &= 0 \qquad\qquad\quad , \text{if idle period,}\\ &= \lambda_{\text{eff}}/[1 - (\alpha + \lambda_{\text{eff}})/(\exp(\alpha + \lambda_{\text{eff}}) - 1)], \quad \text{if collision,}\end{aligned}$$

where $\lambda_{\text{eff}} = \lambda$ (length of the period).

Then the retransmission probability f is updated as

$$f_{n+1} = \min(\beta, \alpha/\hat{Z}_{n+1}).$$

If the traffic intensity λ is not known then an estimate of λ is also provided in Thomopoulos (1988). Under these estimates the following result for stability has been obtained. The packet length is taken as 1 slot.

COROLLARY.

The maximum stable throughput obtained by 1-persistent CSMA, under the above adaptive scheme, is the solution of

$$(1 - x)e^{-x} = 1/(1 + \tau),$$

where τ is the time required by the stations to sense the channel being idle. For $\tau \ll 1$, $\lambda_{\max} \approx 1 - \sqrt{2\tau}$. For CSMA/CD the maximum stable throughput is obtained by solving the equation

$$(1 - x)e^{-x} = 1/(1 + \tau/\tau_c),$$

where τ_c is the fraction of a slot used in collision detection.

Thomopoulos (1988) defines the stability in the following way: the average rate of successful transmission should be maintained and the expected delay should remain bounded as the number of users sharing the channel approaches infinity.

The control scheme used in Cunningham & Medith (1988) applies stochastic approximation techniques, following Merakos & Kazakos (1985) and Hajek (1982, 1985).

6. Polling schemes

In this scheme, the different users sharing the channel are polled in a certain order for using the channel. The polling in a physical system can be done in various ways. One common method is by sending a token to different stations whenever the channel is free. When a station/user is polled, it may or may not be allowed to send more than one packet/message at a time. Even TDMA can be considered a polling system.

But in general, in a polling system the disadvantage of TDMA, in having the channel idle for a long time when there are a large number of users, is not there. Therefore, these schemes can be very efficient in local area networks (Sachs 1988; Bertsekas & Gallagher 1987). Hence, polling systems have been extensively studied. Since rather exhaustive surveys are available for these systems (see Takagi 1986, 1990) we shall limit ourselves to only the references which are directly relevant for the stability of the polling systems.

The polling systems have been studied under various assumptions. The number of stations $M < \infty$. The stations can have finite or infinite buffers. The time axis can be slotted or unslotted. The number of messages allowed for transmission at a time from a user can be one, $k \geqslant 1$, exhaustive (i.e. all packets queued at the station) or gated (with various restrictions). The walk time, i.e., the time to poll from one user to another or the time a token takes to reach from one user to another can be zero, deterministic or random. If the walk time is zero then the stability of polling systems can be studied in the generality of § 3. If walk time is nonzero then the stationary distributions in such generality may not exist. Therefore, in this section we consider the stability of polling systems where each user has an infinite buffer (stability problem does not arise for the finite buffer case), the walk times are nonzero and the message generation sequences are Poisson processes (for the unslotted case) or i.i.d. sequences (for the slotted case).

The main references for stability of the polling systems are Eisenberg (1972), Hashida (1972), Konheim & Meister (1974), Zhdanov & Saksonov (1979) and Swartz (1980). A nice survey emphasising the performance and technological aspects of token rings is given is Bux (1989). In Konheim & Meister (1974) and Swartz (1980) the time axis is considered slotted with i.i.d. input sequences while in Eisenberg (1972), Hashida (1972) and Zhdanov & Saksonov (1979) the system is unslotted. The arrival processes at each user are Poisson in Hashida (1972) and Eisenberg (1972), while more general renewal arrival process at each user is considered in Zhdanov & Saksonov (1979). But the criterion of stability, which is ergodicity of appropriate Markov chains in Hashida (1972) and Eisenberg (1972), is the existence of finite limit of expected values of certain system parameters (for Poisson input the results in Zhdanov & Saksonov (1979) reduce to that of others).

We present, in the following, the results of Zhdanov & Saksonov (1979). Eisenberg (1972), Hashida (1972), Konheim & Meister (1974) and Swartz (1980) have used Laplace–Stieltjes transform techniques, which involve many tedious manipulations. Also, as mentioned above, Zhdanov & Saksonov (1979) do convey something in the case of general renewal input process also.

There are $1 < M < \infty$ users each with an infinite buffer. The input messages generated at the users are independent renewal processes. The interarrival time at user i has mean $0 < \alpha_i < \infty$ and the service time has general distribution with mean $0 < \beta_i$. Walk time from user i to user $i+1$ is random having a general distribution with finite mean $0 < \gamma_i < \infty$. All random variables are independent of each other. We consider in the following only the gated system although the exhaustive service system has also been treated by Zhdanov & Saksonov (1979).

For the system described above, it is easy to show that

$$t_{ik} = \left[\sum_{j=1}^{M} \gamma_j\right] \Big/ \alpha_i + \left[\sum_{j=1}^{i-1} t_{jk}\beta_j + \sum_{j=i+1}^{M} t_{jk-1}\beta_j\right] \Big/ \alpha_i,$$

where t_{ik} is the expected value of the total time the server spends at user i in the kth cycle (i.e., the kth time it comes to user i). This can be rewritten as

$$t_k = (I - A)^{-1} B t_{k-1} + (I - A)^{-1} \gamma,$$

where,

$$t_k = (t_{1k}, t_{2k}, \ldots, t_{Mk})^T,$$

$$\gamma = \left(\sum_{i=1}^{M} \gamma_i \right)^{-1} (\beta_1/\alpha_1, \beta_2/\alpha_2, \ldots, \beta_M/\alpha_M)^T,$$

$$A = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \rho_2 & 0 & \cdots & 0 \\ \rho_3 & \rho_3 & 0 & 0 \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ \rho_M & \rho_M & \cdots & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} \rho_1 & \rho_1 & \rho_1 & \cdots & \rho_1 \\ 0 & \rho_2 & \rho_2 & \cdots & \rho_2 \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ 0 & 0 & 0 & \cdots & \rho_N \end{bmatrix},$$

and

$$\rho_i = \beta_i/\alpha_i.$$

Now, it is well-known that this recursive equation has a finite limiting value if and only if $(I - A)^{-1}B$ has all its eigenvalues in a unit circle in the complex plane. It has been shown in Zhdanov & Saksonov (1979) that a sufficient condition for this is

$$\sum_{i=1}^{M} (\beta_i/\alpha_i) < 1. \tag{2}$$

The finiteness of the limiting value of t_{ik} implies the finiteness of the limiting value of expected valued η_{ik} (as $k \to \infty$) of the number of customers served in kth cycle at user i.

If the arrival processes at the users are independent Poisson processes then the above result gives the ergodicity of the Markov chain of queue length processes formed at the instants of arrival of the server at ith user. Analysis in the following differs somewhat from that in Zhdanov & Saksonov (1979) and provides a little stronger result which is equivalent to that obtained in Hashida (1972) and Eisenberg (1972). We denote $(z_n(1), z_n(2), \ldots, z_n(M))$ by z_n, where $z_n(j)$ is the queue length at the jth user when the server arrives at the ith queue (fixed in the following) in the nth cycle. Then $\{z_n\}$ is an aperiodic, irreducible Markov chain. We show in the following that the above result on finiteness of the limit of the expected values implies that $\{z_n\}$ is ergodic.

It can be easily seen that $z_n(j)$, $j \neq i$ is a.s. dominated by the queue length at user j when the server visits j either in nth cycle or in $(n+1)$th cycle. Thus $\lim_{n\to\infty} \sup E[z_n(j)] < \infty$. Now we define a random sequence $\{s_n\}$ by

$$s_n = \sum_{j=1}^{M} z_n(j).$$

Then $Es_n = \Sigma_{j=1}^{M} E[z_n(j)]$ and $\lim_{n\to\infty} \sup E(s_n) < \infty$. Since $\{z_n\}$ is irreducible, if $\{z_n\}$ is not an ergodic sequence then $P\{z_n = a\} \to 0$ for any vector $a = (a_1, \ldots, a_M)$. This implies that $P\{S_n = k\} \to 0$ for all $k \geqslant 0$. Therefore,

$$\liminf_{n\to\infty} Es_n = \liminf_{n\to\infty} \sum_{k=0}^{\infty} P\{S_n \geqslant k\} \geqslant \sum_{k=0}^{\infty} \lim_{n\to\infty} P\{S_n \geqslant k\} = \infty.$$

Hence $\{z_n\}$ is an ergodic Markov chain.

We now mention a few more papers on the stability of the polling systems. The stability in these papers may not imply the ergodicity of any underlying Markov chain. An influential paper is by Kuehn (1979). Analysis of this kind has been followed up in Servi (1985) and Ibe & Chang (1989). See Takagi (1990) for many more references.

7. Collision resolution algorithms

We have seen in the earlier sections that for infinite user models, slotted ALOHA, CSMA and CSMA/CD are all unstable for any nonzero input traffic. Collision resolution algorithms (CRA) introduced in Capetanakis (1979) and Tsybakov & Mikhailov (1978) were the first algorithms among the random multiple access methods which provide nonzero stable throughput. The currently available CRA have throughputs upto 0·487 which exceed the throughput of $1/e$ available from controlled ALOHA. The IEEE special issue (*IEEE Trans. Inf. Theory* 1985) contains the history, most of the references to earlier literature and the state of the art till 1984 on these algorithms. We shall mention some more recent work at the end of this section.

The number of users $M = \infty$ and each user has a single buffer. The system is slotted and the packet lengths are fixed. Total system packet generation process is Poisson. A nonblocked binary stack algorithm is described as follows (Tsybakov 1985).

(i) If a station has a packet to transmit, then in the time interval from the packet's generation to its successful transmission, it assigns to this packet an integer L_n in slot n. L_n is called its packet level at time n. The packet is transmitted over the channel in slot n if and only if $L_n = 0$.
(ii) Every new packet is assigned level 0.
(iii) If $L_n = 0$ for a packet and there is collision in slot n then $L_{n+1} = 1$ with probability $1/2$, and $L_{n+1} = 0$ otherwise.
(iv) If $L_n \geqslant 1$ for a packet and there is a collision in slot n then $L_{n+1} = L_n + 1$.
(v) If $L_n \geqslant 1$ and there is no collision in slot n then $L_{n+1} = L_n - 1$.

A generalization of the above algorithm is nonblocked J-ary stack algorithm in which the steps (iii) and (iv) in the above algorithm are changed to:

(iii)′ If $L_n = 0$ for a packet and there is a collision in slot n then L_{n+1} takes value j with probability P_j, where $j \in \{0, 1, \ldots, J-1\}$.
(iv)′ If $L_n \geqslant 1$ and there is a collision in slot n then

$$L_{n+1} = L_n + J - 1.$$

The above algorithm has been modified to eliminate certain predictable collisions. The details are in Massey (1981, pp. 73–137). In another algorithm called blocked stack algorithm (Tsybakov & Mikhailov 1978), the newly generated packets are not

allowed to transmit immediately. If there has been a collision in the recent past, all the packets involved in that collision are first allowed to transmit. Only after their successful transmission, packets generated during that time transmit and then if there is a collision, these packets get into what is called a collision resolution period. The disadvantage of the blocked algorithm as compared to the nonblocked one, explained above, is that in the blocked case all the stations have to keep monitoring the channel throughout whether they have a packet to transmit or not. A further variation along the theme of collision resolution is the part and try algorithm (Tsybakov & Mihailov 1980).

In the following we provide the stability results for the blocked stack algorithm. The analysis for the nonblocked is along the same lines. Assume that x_k is the number of new packets generated in the system in the kth slot. The sequence $\{x_k\}$ is assumed i.i.d. Also, let

$$P\{x_k = m\} = \lambda^m e^{-\lambda}/m!, \quad m = 0, 1, \ldots$$

The analysis can easily be extended to the general distributions. Let us denote by τ_k the length of a collision resolution period given that the collision multiplicity (number of packets involved in the initial collision) is k. Then, the following recursive equation holds.

$$E[\tau_k] = P_0(1 + E(\tau_k)) + \sum_{n=1}^{k} P_n(2 + E(\tau_{k-n}) + E(\tau_n)),$$

where $P_l = 2^{-k}\binom{k}{l}$ and the initial conditions are $E[\tau_0] = E[\tau_1] = 0$. This provides $E[\tau_2] = 3{\cdot}5$, $E[\tau_3] = 6$ etc. Also, it can be shown (Tsybakov & Mikhailov 1978) that

$$[(8/3) - (1/168)]k - 2 \leqslant E[\tau_k] \leqslant (8/3)k - 2. \tag{3}$$

We denote by ξ_n the multiplicity of the nth conflict. Under the assumption of $\{x_n\}$ being i.i.d., it can be shown that $\{\xi_n\}$, and $\{\tau_n\}$ are homogeneous Markov chains. Using the results on the ergodicity of Markov chains similar to theorem 1 and lemma 2, the following theorem can be obtained.

Theorem 4. *Under the above assumptions the following hold (Tsybakov & Mikhailov 1978)*

(i) *If* $\lambda < \liminf_{k\to\infty}(k/E\tau_k)$ *then* $\{\xi_n\}$ *is an ergodic Markov chain.*
(ii) *If* $\lambda \leqslant \liminf_{k\to\infty} k/E\tau_k$ *then* $\{\xi_n\}$ *is a recurrent Markov chain.*
(iii) *If* $\lambda > \limsup_{k\to\infty} k/E\tau_k$ *then* $\{\xi_n\}$ *is a transient Markov chain.*

From the estimates given in (3), theorem 4 states that if $\lambda < 3/8$ then $\{\xi_n\}$ is ergodic and if $\lambda \geqslant 3/8$ then it is recurrent. It has been shown (Tsybakov & Mikhailov 1978) that if $\lambda < 3/8$ then $\{\tau_n\}$ is also an ergodic Markov chain, that $\lim E\tau_n < \infty$ and that $\lim E\xi_n < \infty$.

Recently sharper asymptotic estimates have been obtained for the $E[\tau_k{}^m]$, for $m \geqslant 1$ (Jacquet & Szpankowski 1989). The performance of the collision resolution algorithms can be improved somewhat if the channel feedback is used to estimate the collision multiplicity also (Cidon & Sidi 1986). This estimate is then used to choose dynamically the J-parameter in a J-ary stack algorithm.

In a part and try algorithm (Tsybakov & Mihailov 1980; Gallagher 1978) the

collision resolution is done by allowing the colliding packets to transmit in order of their time of generation. This is done by splitting the time axis and allowing packets to be transmitted if they arrived in a certain subset of the time axis. In Mosely & Humblet (1985) and Khasminskii (1989) the problem of splitting the time axis optimally has been addressed by forming a Markov decision problem.

The collision resolution algorithms discussed above have been in the framework when the channel propagation time is large. In local area networks (LAN), collisions and carriers can be sensed in a short time and, exploiting this feature, collision resolution algorithms for LAN have been studied in Georgiopoulos *et al.* (1986, pp. 185–214).

8. Hybrid techniques

As described in § 3, random access schemes perform well when there are a large number of bursty users with light traffic and reservation based schemes like TDMA provide good performance when the number of users is small and traffic intensity is high. This has motivated many researchers to develop hybrid schemes combining random access and reservation techniques. Our scheme in § 3 is motivated by the same idea. Several references to the hybrid schemes are provided in Sachs (1988). Also, several hybrid schemes are mentioned in Mehravari (1984) and it has been shown that under heavy load the TDMA is better than any random access scheme. Multiple access schemes with priority have been reviewed in de Moraes (1990).

An error in the first draft of this paper was pointed out by the reviewer.

References

Abramson N 1970 The ALOHA system – Another alternative for computer communications. In *Proc. Fall Joint Computer Conf.* (Montvale, NJ: AFIPS Press) vol. 37, pp. 281–285

Aldous D J 1987 Ultimate instability of exponential back-off protocol for acknowledgement based transmission control of random access communication channels. *IEEE Trans. Inf. Theory* IT-33: 219–223

Bambos N, Walrand J 1989 On queues with periodic inputs. *J. Appl. Probab.* 26: 381–389

Bertsekas D, Gallagher R 1987 *Data networks* (Englewood Cliffs, NJ: Prentice-Hall)

Beuerman S L, Coyle E J 1988 The delay characteristics of CSMA/CD networks. *IEEE Trans. Commun.* 36: 553–563

Bux W 1989 Token ring local area networks and their performance. *Proc. IEEE* 77: 238–256

Capetanakis J I 1979 Tree algorithm for packet broadcast channels. *IEEE Trans. Inf. Theory* IT-25: 505–515

Cidon I, Sidi M 1986 Conflict multiplicity estimation and batch resolution algorithms. *INFOCOM'86*: (New York: IEEE Press) pp. 197–203

Clare L P 1986 Delay analysis of stable slotted ALOHA systems. *INFOCOM'86* (New York: IEEE Press) pp. 10–19

Cottrell M, Claude Fort J, Malgouyres G 1983 Large deviations and rare events in the study of stochastic algorithms. *IEEE Trans. Autom. Control* AC–28: 907–920

Coyle E J, Liu B 1985 A matrix representation of CSMA/CD networks. *IEEE Trans. Commun.* COM-33: 53–64

Cunningham G A, Medith J S 1988 Distributed retransmission controls for slotted nonpersistent and virtual time CSMA. *IEEE Trans. Commun.* 36: 685–691

de Moraes L F M 1990 Priority scheduling in multiple access communication. In *stochastic analysis of computer & communication systems* (ed.) H Takagi (Amsterdam: Elsevier Science Publishers)

Eisenberg M 1972 Queues with periodic service and changeover time. *Oper. Res.* 20: 440–451

Gallagher R G 1978 Conflict resolution in random access broadcast networks. In *Proc. AFOSR Workshop on Commun. Theory Appl.* pp. 74–76

Georgiopoulos M, Merakos L, Papatani-Kazakos P 1986 High performance asynchronous limited sensing algorithms for CSMA and CSMA/CD channels. In *Local area and multiple access networks* (ed.) R L Pickholtz (Rockville, MD: Computer Science Press) pp. 185–214

Ghez S, Verdu S, Schwartz S C 1989 Optimal decentralized control in the random access multipacket channel *IEEE Trans. Autom. Control* 34: 1153–1163

Goodman J, Greenberg A G, Madras N, March P 1988 Stability of binary exponential backoff. *J. Assoc. Comput. Mach.* 35: 579–602

Hajek B 1982 Hitting time and occupation time bounds implied by drift analysis with applications. *Adv. Appl. Probab.* 14: 502–525

Hajek B 1985 Stochastic approximation methods for decentralized control of multiaccess communications. *IEEE Trans. Inf. Theory* IT-31: 176–184

Hashida O 1972 Analysis of multiqueue. *Rev. Elec. Commun. Lab.* 20: 189–199

Hastad J, Leighton T, Rogoff B 1987 Analysis of backoff protocols for multiple access channels. *Proc. 19th Annu. ACM Symp. on Theory of Comput.* pp. 241–243

Ibe O C, Chang X 1989 Stability conditions for multiqueue systems with cyclic service. *IEEE Trans. Autom. Control* 33: 102–103

IEEE Trans. Inf. Theory 1985 vol. IT-31, Special issue on Random-Access Communications

Jacquet P, Szpankowski W 1989 Ultimate characteristics of interval searching algorithms for distributed systems. In *Performance of distributed and parallel systems* (eds) T Hasegawa, H Takagi, Y Takahashi (Amsterdam: Elsevier Science Publishers)

Khasminskii R Z 1989 Some optimal algorithms of random multiple access. *Theory Probab. Appl.* 34: 454–463

Kleinrock L, Tobagi F A 1975 Packet switching in radio channels: part I – Carrier sense multiple access models and their throughput – delay characteristics. *IEEE Trans. Commun.* Com-23: 1400–1416

Konheim A G, Meister B 1974 Waiting lines and times in a system with polling. *J. Assoc. Comput. Mach.* 21: 470–490

Kuehn P J 1979 Multiqueue systems with nonexhaustic cyclic service. *Bell Syst. Tech. J.* 58: 671–698

Lim J T, Meerkov S M 1987 Theory of Markovian access to collision channels. *IEEE Trans. Commun.* COM-35: 1278–1288

Loynes R 1962 The stability of a queue with non-independent interarrival and service times. *Proc. Cambridge Philos. Soc.* 58: 497–520

Massey J L 1981 Collision resolution algorithms and random access communications. In *Multiuser communications systems* (ed.) G Longo (CISM courses and lectures No. 265) (New York: Springer)

Medith J S, Lea C-T A 1983 Stability and optimization of the CSMA and CSMA/CD channels. *IEEE Trans. Commun.* COM-31: 763–774

Mehravari N 1984 TDMA in a random access environment, an overview. *IEEE Commun. Mag.* 22(11): 54–59

Merakos L, Kazakos D 1985 On the retransmission control policies in multiple access communication networks. *IEEE Trans. Autom. Control.* AC-30: 109–117

Mosely J, Humblet P A 1985 A class of efficient contention resolution algorithms for multiple access. *IEEE Trans. Commun.* COM-33: 145–151

Nelson R 1984 The stochastic cusp, swallowtail and hyperbolic umbilic catastrophes as manifest in a simple communications model. *Performance'84* (Amsterdam: Elsevier Science Publishers)

Neuts M F 1981 Matrix geometric solutions in stochastic models (Baltimore: Johns Hopkins Univ. Press)

Onozato Y, Noguchi S 1985 On the thrashing cusp in slotted ALOHA systems. *IEEE Trans. Commun.* COM-33: 1171–1182

Pakes A G 1969 Some conditions for ergodicity and recurrence of Markov chains. *Oper. Res.* 117: 1059–1061

Paterakis M, Georgiadis L, Papatani-Kazakos P 1987 On the relation between the finite and the infinite population models for a class of RAA's. *IEEE Trans. Commun.* COM-35: 1239–1240

Prathima B V 1991 *Stability and performance analysis of same multiple access and mobile cellular communication networks*, MS thesis, Department of Electrical Engineering, Indian Institute of Science, Bangalore

Rivest R L 1987 Network protocol by Bayesian broadcast. *IEEE Trans. Inf. Theory* IT-33: 323–328

Rolski T 1981 *Stationary random processes associated with point processes* (New York: Springer-Verlag)

Rosenkrantz W A, Towsley D 1983 On the instability of the slotted ALOHA multiaccess algorithm. *IEEE Trans. Autom. Control* AC-28: 994–996

Rubin I 1979 Message delay in FDMA and TDMA communication channels. *IEEE Trans. Commun.* COM-27: 516–536

Rubin I 1983 Synchronous and channel sense asynchronous dynamic group-random-access scheme for multiple access communications. *IEEE Trans. Commun.* COM-31: 1063–1076

Saadawi T N, Ephremides A 1981 Analysis, stability and optimization of slotted ALOHA with a finite number of buffered users. *IEEE Trans. Autom. Control.* 26: 680–689

Sachs S R 1988 Alternative local area network access protocols. *IEEE Commun. Mag.* 26(3): 25–44

Servi L D 1985 Capacity estimation of cyclic queues. *IEEE Trans. Commun.* COM-33: 279–282

Sharma V 1987 Stability analysis of slotted ALOHA with time varying input, NASA Tech. Report

Sharma V 1988 Stability and continuity for slotted ALOHA with stationary non-independent input traffic. *J. Appl. Probab.* 25: 797–807

Swartz G B 1980 Polling in a loop system. *J. Assoc. Comput. Mach.* 27: 42–59

Szpankowski W 1986 Stability, bounds and approximation in asymmetric buffered ALOHA-type systems. *Proc. 25th Conf. Decision and Control* (New York: IEEE Press) pp. 1722–1727

Takagi H 1986 *Analysis of polling systems* (Cambridge, MA: MIT Press)

Takagi H 1990 Queueing analysis of polling models: An update. In *Stochastic analysis of computer and communication systems* (ed.) H Takagi (Amsterdam: Elsevier Science Publishers)

Thomopoulos S C A 1988 A simple and versatile decentralized control for slotted ALOHA, reservation ALOHA and local area networks. *IEEE Trans. Commun.* 36: 662–674

Thomopoulos S C A, Sunder R 1987 Delay analysis of the controlled ALOHA: exact analysis and bounds. *IEEE INFOCOM'87* (New York: IEEE Press) pp. 26–33

Tobagi F A 1982 Distributions of packet delay and interdeparture time in slotted ALOHA and carrier sense multiple access. *J. Assoc. Comput. Mach.* 29: 907–922

Tobagi F A, Hunt V B 1987 Performance analysis of carrier sense multiple access with collision detection. In *Advances in local area networks* (ed.) Karl Kummerle, J O Limb, F A Tobagi (New York: IEEE Press)

Tobagi F, Kleinrock L 1977 Packet switching in radio networks, Part IV. Stability considerations and dynamic control in carrier sense multiple access. *IEEE Trans. Commun.* COM-25: 1103–1120

Tsitsklis J N 1987 Analysis of a multiaccess control scheme. *IEEE Trans. Autom. Control* AC-32: 1017–1029

Tsybakov B S 1985 Survey of USSR contributions to random multiple access communications. *IEEE Trans. Inf. Theory* IT-31: 143–165

Tsybakov B S, Mikhailov V A 1978 Free synchronous access of packets to a broadcast channel with feedback. *Probl. Peredachi Inf.* 14(4): 32–59

Tsybakov B S, Mikhailov V A 1979 Ergodicity of ALOHA system. *Probl. Peredachi Inf.* 15: 75–87

Tsybakov B S, Mihailov V A 1980 Random multiple access of packets: part and try algorithm. *Probl. Peredachi Inf.* 16: 65–79

Zhdanov V S, Saksonov E A 1979 Conditions of existence of steady state modes in cyclic queueing systems. *Avtom. Telemekh.* (Engl. Transl.) 40: 176–184

Task allocation in multiserver systems – A survey of results

ANURAG KUMAR

Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India

Abstract. Jobs consisting of one or more tasks arrive to a system comprising several servers, each with its own queue. Each task requires a single service at any of the servers, and a job completes service when all its constituent tasks have been serviced. Such models arise in the performance modelling of distributed computing systems, computer communication networks, and manufacturing systems. We survey the literature on this class of models.

We classify jobs as being of one of three types: single tasks, multitask with precedence constraints, and multitask stream jobs. After surveying the optimal allocation problem for single task jobs, we discuss the results on the performance analysis and optimal allocation of tasks for the other two job types.

Keywords. Control of queues; customer allocation; routing; load balancing.

1. Introduction

This paper is a summary of published, and a few to be published, results pertaining to customer allocation in single station queueing systems with multiple servers. Customers or jobs, each comprising one or more tasks, arrive to the system. Each task requires a single service from a server, after receiving which, it departs. The service of a job is complete after all its constituent tasks have been served. Although servers may have different speeds, each server can serve any task, and tasks do not inherently have any preference to be served at any particular server. We study the effect on job sojourn time of various policies for allocating tasks to servers.

Figure 1 is a schematic representation of the general model described above. Our description has left unspecified the number and location of the queues, the allocation strategy of customers to queues, and the strategy for allocating servers to service the queued customers. Thus this general model gives rise to several subclasses of models depending on the choices we make among one of the many alternatives we have in the general model.

1.1 *Job type*

Each job may consist of a *single task*. Each task is given one indivisible/uninterruptible service by a server. After completion of this one task the job departs. In this case the job sojourn time is the sojourn time of the task. This is the classical problem, and

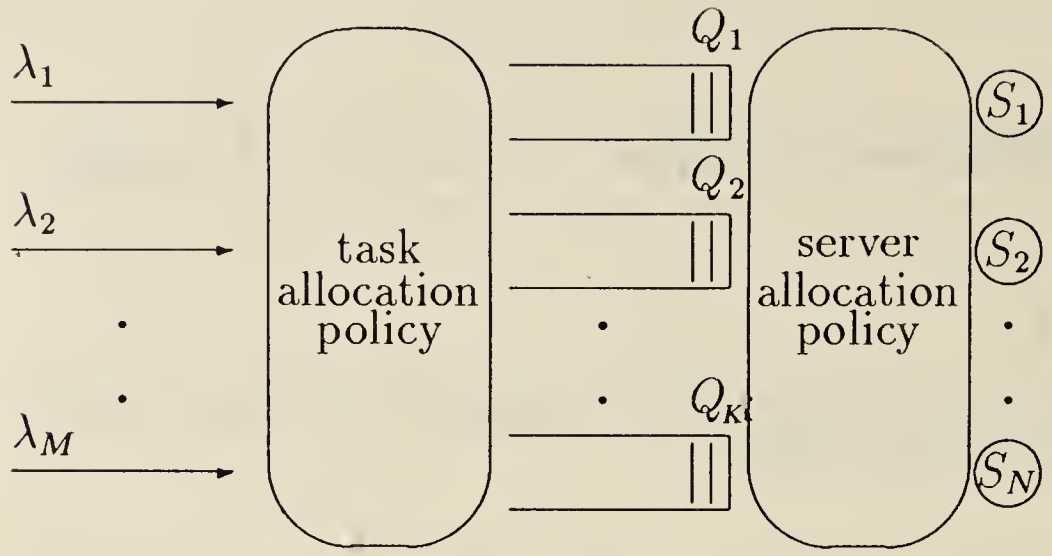


Figure 1. Schematic of the general problem of task allocation to multiple servers.

has received much attention in the literature. Alternatively, each job may comprise *several tasks* (in general, a random number of tasks) that have some precedence relation between themselves. A job is completed when all the tasks in it have been completed. All the tasks may arrive together, in which case the job sojourn time is the time between the arrival epoch of the job and the departure epoch of the last task. Such a model arises in parallel processing and manufacturing systems (Baccelli & Makowski 1990). Lastly, the tasks in a job may arrive in a stream, and must depart in the same order in which they arrived. Here we will only be concerned with the task sojourn times, since job sojourn time has no useful meaning in this case. Such jobs arise as models of virtual circuits in a computer communication network (Jean-Marie 1988, pp. 75–88; Kuri 1990).

1.2 *Queueing and service strategy*

The number of queues K is, in general, less than, equal to, or greater than M. The servers may be *statically allocated* to queues, i.e., each server serves customers from only one queue. Alternatively, the servers may not be attached to queues, but there is a *static policy* for the allocation of servers to queues, e.g., cyclic service. Finally the assignment of servers to queues may be *dynamic*, in the sense that system state information may be used to decide which queue(s) to serve next (Lin & Kumar 1987).

The focus of this paper is on the class of models depicted in figure 2. Server $i, 1 \leqslant i \leqslant N$, is statically assigned to queue $i, 1 \leqslant i \leqslant N$. This is a commonly occurring model, and our own work on this model has been motivated by models for multicomputer systems, and of communication links emanating from a packet switch. In the latter case a "job" corresponds to a virtual circuit which is a stream of packets (tasks).

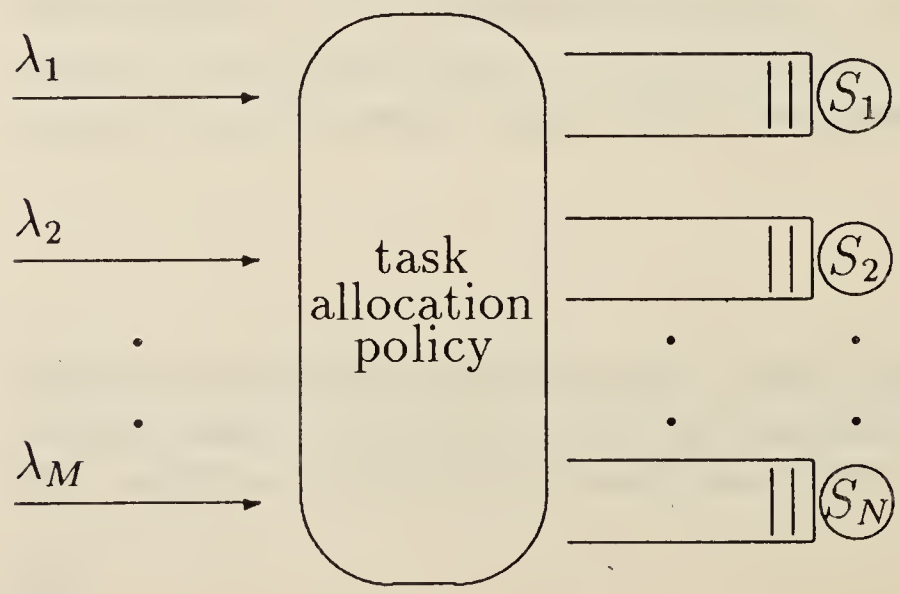


Figure 2. Schematic of the class of problems surveyed in this paper.

1.3 *Task allocation strategy*

The task allocation strategy may be *static*, which means that no system state information, nor any record of past allocations, is used to decide which queue to allocate a task to. Probabilistic allocation is a static allocation strategy. *Semidynamic* allocation strategies do not use dynamic information of system state, but do keep track of past allocations to decide where to put the next task (Agrawala & Tripathi 1981; Yum 1981). Cyclic (or round-robin) allocation belongs to this class of strategies. Finally, a *dynamic* allocation strategy uses system state information when allocating a task. Information that may be used includes queue lengths, work in the system, residual service (Weber 1978; Ephremides *et al* 1980).

In this paper we will consider all the types of task allocation strategies discussed above. We will, however, not permit "jockeying" and preemption. Tasks will be irrevocably assigned to a queue, and will be served to completion, without preemption, by the server at that queue.

The rest of the paper is organised as follow. In § 2 we discuss the classical problem in which each job has a single task. In § 3 we present the more recent results for the class problems with several tasks per job, all of which arrive together. In the context of a continuous arrival stream of such jobs, the only precedence constraints that seem to have been considered in the literature are the fork-join constraints, that give rise to the class of fork-join queueing models. In § 4 we turn our attention to jobs that consist of a stream of tasks that must depart in sequence. We conclude in § 5 with a discussion of some problems for further research.

2. Single task jobs

There are N servers, each with its own queue. Several job streams arrive to the system. Some of the streams may be dedicated to certain servers and others may be allocable. This class of models also arises in the study of load sharing (or load balancing) in computer systems. We begin by assuming that the servers have the same service rates.

We start with a fundamental result that bounds the performance of this class of models.

2.1 *A general lower bound*

If the service times across all the arriving jobs are independent and identically distributed (i.i.d.), then it is proved in Wolff (1977) that the system shown in figure 3

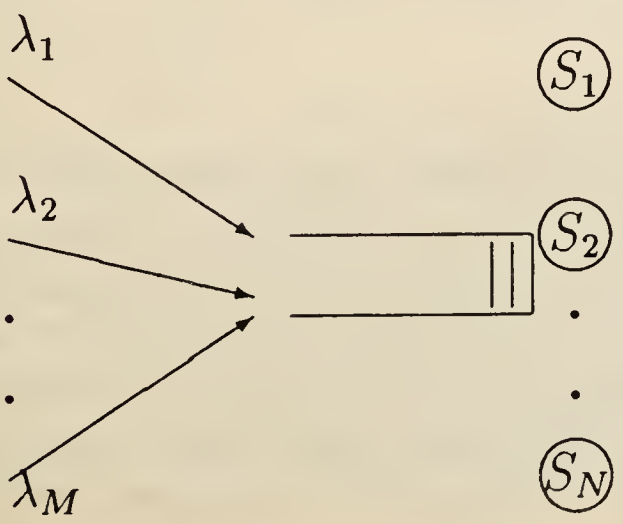


Figure 3. The lower bound system (G/GI/N).

(the G/GI/N system) is a lower bound to any system in the class shown in figures 1, or 2, in the sense of the following theorem

Define $Q(t)$ = queue length at time t for the G/GI/N system and $Q'(t)$ = queue length at time t for any other queueing and allocation policy.

Theorem 1. *For all* $t \geqslant 0$, *and for all* $k \geqslant 0$, $P(Q(t) \geqslant k) \leqslant P(Q'(t) \geqslant k)$, *i.e.,* $Q'(t) \geqslant_{st} Q(t)$, *where* $\geqslant_{st}$ *is read as "stochastically greater than or equal to" (i.e., the stochastic ordering).*

Proof. The proof (Wolff 1977) uses the coupling argument (see Kemae *et al* 1977, Smith & Whitt 1981; Ross 1983). Two new queueing processes with queue lengths $\tilde{Q}(t)$ and $\tilde{Q}'(t)$ are constructed by sampling the service times of tasks as they enter service. Thus on each sample path ω, the nth task to enter service in both systems has the same service time. It is shown that $\tilde{Q}(t)(\omega) \leqslant \tilde{Q}'(t)(\omega)$ for all $t \geqslant 0$, and all ω. Owing to i.i.d. service times (and the task selection policy not depending on the service times of queued customers) $\{\tilde{Q}(t), t \geqslant 0\}$ has the same law as $\{Q(t, t \geqslant 0\}$, and $\{\tilde{Q}'(t), t \geqslant 0\}$ has the same law as $\{Q'(t), t \geqslant 0\}$. This completes the coupling argument and the stochastic comparison holds. □

If a stationary queue length distribution exists in each case, with EQ and EQ' denoting the limiting time average queue lengths, it follows by Little's Law that $EW \leqslant EW'$.

It is intuitively clear, and this is also the key point in the proof in Wolff (1977), that the G/GI/N system yields the best queueing performance since it does not allow a situation where there is a task waiting at one queue and an idle server is available at another queue. In a typical distributed system (one of the motivations for the class of models under study here), owing to communication delays, the G/GI/N model is inapplicable since this would require that the idleness of a server be communicated to all the queues instantaneously. Recall, further, that in this survey we are only considering models in which tasks are irrevocably assigned to queues, i.e., there is no "jockeying". Models that allow task migration have been studied by Eager *et al* (1986) and Mirchandaney *et al* (1989). If communication delays are small compared to task service times then it may be possible to approach the G/GI/N performance.

That the policy for allocating tasks to queues, as the tasks arrive, can have a significant effect on delay performance is demonstrated by the following simple example. Consider the model in figure 2 with $N = 2$ and a single Poisson arrival process of tasks, with rate λ, each requiring service with an exponential distribution with mean μ^{-1}, at either server. Assume that $\rho = \lambda/2\mu = 0{\cdot}9$, and $\mu = 1$. The M/M/2 ideal yields a mean task sojourn time of 4·26, whereas cyclic (alternating) allocation yields 6·81, and random allocation, with probability $\frac{1}{2}$ to each server, yields 9·00.

2.2 *Dynamic policies*

The most well-known references (Winston 1977; Weber 1978; Ephremides 1980) concerning dynamic policies all conclude that, by one criterion or another, the "Join-the-Shortest-Queue" discipline is optimal. We discuss the result of Weber (1978).

There are N identical servers, the tasks arrive according to an arbitrary process and have i.i.d. service times with non-decreasing hazard rate, i.e., if B denotes the service time random variable, $B(\cdot)$ denotes the distribution and $b(\cdot)$ the density, then $b(t)/(1 - B(t))$ is nondecreasing in $t, t \geqslant 0$. An arriving task can observe the state

$(Q_1, Q_2, \ldots, Q_N; X_1, \ldots, X_N)$, where $Q_i, 1 \leqslant i \leqslant N$, is the number of customers waiting in queue i, and X_i is the amount of service already rendered to the customer in service, with $X_i = 0$ if the server i is idle. The task allocation policy S^* is such that if an arriving task finds the state $(q_1, q_2, \ldots, q_N; x_1, \ldots, x_N)$ then S^* assigns the task to queue k, where $k = \arg\min_{1 \leqslant i \leqslant N}(q_i EB + E(B - x_i | B > x_i))$, i.e., the policy assigns the task to the queue with the least expected work. Let $D(t)$ be the departure process of the system. Denoting by $P^*_{(\mathbf{q}_0, \mathbf{x}_0)}$ (respectively $P_{(\mathbf{q}_0, \mathbf{x}_0)}$) the probability measures under policy S^* (respectively S) and initial state $(\mathbf{q}_0, \mathbf{x}_0)$, the following result is obtained.

Theorem 2. *$P^*_{(\mathbf{q}_0, \mathbf{x}_0)}(D(t) \geqslant k) \geqslant P_{(\mathbf{q}_0, \mathbf{x}_0)}(D(t) \geqslant k)$, i.e., the number of departures till time t is stochastically greater under S^* than under S.*

Proof. See Weber (1978). □

Letting $M(t)$ denote the number of tasks in the *system* at time t, and observing that $M(t) = M(0) + A(t) - D(t)$ where $A(t)$ is the arrival process, we have the following result.

COROLLARY.

(i) *$P^*_{(\mathbf{q}_0, \mathbf{x}_0)}(M(t) \geqslant k) \leqslant P_{(\mathbf{q}_0, \mathbf{x}_0)}(M(t) \geqslant k)$ i.e., the number in the system at time t is stochastically smaller with S^* than with any other policy.*
(ii) *If limiting distributions exist for $M(t)$ under each policy and are denoted M^* and M, then $M^* \leqslant_{st} M$, where $\leqslant_{st}$ reads "stochastically less than or equal to".*
(iii) *If ET^* and ET denote the mean task sojourn times then $ET^* \leqslant ET$.*

Proof. (i) is obvious, and (ii) follows from the weak convergence property of stochastic ordering (Stoyan 1983); (iii) is an immediate consequence of Little's Law. □

Observe that when the system is restricted to exponential service times (but an arbitrary arrival process) then the policy S^* becomes the join-the-shortest-queue (JSQ) policy. Winston (1977), assuming Poisson arrivals and exponential service times, shows that JSQ is optimal in the sense of stochastically maximising the discounted number of departures till time t(i.e., $\int_0^t e^{-\alpha u}\, dD(t)$ where $\alpha > 0$). For the special case of two queues, Ephremides *et al* (1980) assume exponential service times, and arbitrary arrivals, to show that JSQ is optimal for minimising the expected time till the system becomes empty, if the arrival stream is stopped at some time t. The latter paper also considers an important variation on the information structure of the problem. It is shown that if queue lengths are not observable but are known to be equal at the beginning, then the policy of alternately assigning arrivals to queues is optimal for minimising the same cost function.

Whitt (1986) provides a counterexample that shows that the nondecreasing failure rate condition in theorem 2 is necessary. Whitt considers a service time distribution that has a point mass of $1 - \varepsilon$ at 0 and a point mass of ε at n, where ε is small and n is large. This distribution clearly does not have nondecreasing failure rate (see Wolff 1989, p. 480). Whitt argues that for this service time distribution, a policy which follows JSQ if either queue is empty or if the queue length difference is less than or equal to 1 but assigns the task to the longer queue if both queues are nonempty and the difference is 2 or more, is better than pure JSQ. Suppose the queues start out

empty and consider the first time that both servers become busy. At this time both servers are serving tasks whose service times are n, and there is one task in each queue. Eventually the task at the head of one of the queues will complete, and with a high probability the queue will empty out. Arrivals will now join this queue until an arrival with nonzero service time occupies the server. The queue length difference, has a high probability now of being more than 1. It is now beneficial for the next arrival to join the longer queue since the task at the head of this queue will finish earlier than the task in the other queue. Since the other tasks queued up in the longer queue will, with a large probability, have zero service time, the new arrival will enter service earlier if it joins the longer queue. Whitt provides the "epsilonics" to tighten these arguments.

All the results discussed so far in this section have been for identical servers. Clearly if there are two queues and one server is much faster than the other, then, until the queue at the faster server builds up beyond a certain level, tasks should not be assigned to the slower server. Hajek (1984) considers a general model of two interacting service stations of which the following model is a special case. Tasks arrive in a Poisson process, and join one of two queues; at the first queue the service time distribution is exponential with rate μ_1 and at the other the distribution is exponential with rate μ_2. Let $Q_1(t)$ and $Q_2(t)$ denote the number of tasks in queue 1 and queue 2. Two cost functions are considered: Discounted weighted queue length average,

$$E\int_0^\infty e^{\alpha t}(c_1 Q_1(t) + c_2 Q_2(t))\,\mathrm{d}t,$$

and long run time average of weighted queue lengths,

$$\lim_{T\to\infty} \frac{1}{T} E \int_0^T ((c_1 Q_1(t) + c_2 Q_2(t))\,\mathrm{d}t,$$

where $\alpha > 0$ is the discount factor, and $c_1 > 0, c_2 > 0$ are cost rates.

In either case it is established that the optimal strategy is of *switch-over type*, as stated in the following theorem.

Theorem 3. *There is a nondecreasing function* $s: \{0, 1, 2, \ldots\} \to \{0, 1, 2, \ldots\}$, *such that the optimal strategy sends a customer arriving at* t *to queue 1 if* $Q_2(t) \geqslant s(Q_1(t))$, *and to queue 2 otherwise.*

Proof. See Walrand (1988). □.

This result demonstrates the form of the optimal policy, but the switching curve is difficult to compute (of course, if $\mu_1 = \mu_2$ then the earlier results show that $s(i) = i, i \geqslant 0$). As a practical heuristic, for the case with nonidentical servers, Yum & Schwartz (1981) discuss the Join-Biased-Queue (JBQ) rule. For example if $\mu_1 > \mu_2$ there is an integer Δ such that an arrival is sent to queue 2 only if $Q_1 > Q_2 + \Delta$. Using Markov chain analysis the optimal Δ is found for several examples.

Another class of results has to do with the heavy traffic behaviour of queueing systems subject to the optimal allocation policies discussed in this section. It is shown that suitably time scaled and normalised versions of the queue length processes converge weakly in heavy traffic to a certain diffusion process (namely, reflected Brownian motion) (see Foschini 1977, Foschini & Salz 1978, and Reiman 1983 for

the original results, and Flores 1990 for a survey). We do not give the results here but merely state them informally. The heavy traffic limit corresponds to considering a sequence of queueing systems (indexed by n) with the difference between the total arrival rate and total service rate going to zero as $O(1/n^{\frac{1}{2}})$. For two queues with renewal inputs and identical servers, and JSQ allocation, the time scaled and normalised queue length processes for the two queues are considered. It is shown that in the heavy traffic limit the difference between the normalised queue lengths goes to zero in probability, and the sum of the normalised queue lengths converges weakly to the same diffusion limit as the GI/GI/2 system. These results suggest that under heavy traffic JSQ tends to equalise queue lengths, and the total number in the system behaves just like the lower bound GI/GI/2 system (see §2·1). These insights into the behaviour of the JSQ policy have been used in Nelson & Philips (1990) to develop simple and accurate approximate analysis techniques for queues with JSQ allocation. The difficulty of exact analysis is well-known (see Flatto & McKean 1977).

2.3 *Semidynamic policies*

These are policies that do not use system state information but do keep a record of their past decisions. Consider the allocation of tasks arriving in a Poisson stream of rate λ, to two queues with identical i.i.d. service times at each. If instantaneous state information is not to be used, two simple allocation policies come to mind. Assign the tasks to each queue with probability $\frac{1}{2}$, or assign the tasks alternately to the two queues. In the former case each queue becomes an M/G/1 queue and in the latter case each queue is an E_2/G/1 queue, and in both cases the mean interarrival time is $2/\lambda$. Owing to the fact that the E_2 distribution is smaller in the convex ordering sense than the exponential distribution with the same mean, it follows (Wolff 1989) that the waiting time with round-robin allocation is smaller in the convex ordering sense. In particular, round-robin allocation gives smaller mean delay.

As has already been discussed in §2·2 it is shown by Ephremides *et al* (1990) that for identical exponential servers, with equal initial queue lengths, alternating allocation is optimal. This result is extended by Walrand (1988) to multiple queues and the cost function $E\{\int_0^\infty e^{-\alpha t}\Sigma_{i=1}^N Q_i(t)\mathrm{d}t\}$.

For servers with unequal service rates it is still an open question as to what the optimal semidynamic allocation policy is. One possibility is to find the best probabilistic allocation policy (this might, for example, yield $u_1 = \frac{2}{3}, u_2 = \frac{1}{3}$, i.e., allocate an arrival to queue 1 with probability $\frac{2}{3}$ and to queue 2 with probability $\frac{1}{3}$) and then implement these allocation ratios cyclically (in the example, route customers according to the pattern $1, 1, 2, 1, 1, 2, 1, 1, 2, \ldots$). It is shown in Agrawala & Tripathi (1981) by a counterexample that this, in general, does not yield an optimal cyclic policy in the sense of minimising the mean task delay.

Hajek (1985) considers the following problem. Tasks arrive as a renewal point process to a queue where they require exponentially distributed i.i.d. service. Only a fraction $0 < p < 1$ has to be admitted into the queue. A selection sequence is a 0–1 valued sequence $\{r_k, k \geqslant 1\}$ such that $\lim_{n\to\infty}(1/n)\Sigma_{k=1}^n r_k = p$, and task k is accepted into the queue if $r_k = 1$. Hajek introduces the regular sequence $r_k^* = [(k+1)p] - [kp]$, where $[\cdot]$ denotes "greatest integer less than or equal to". Observe that for rational $p = m/d$ the sequence has period d and there are exactly m 1's in each period. The following theorem is proved.

Theorem 4. *The selection sequence* $\{r_k^*\}$ *minimises* $\lim_{n\to\infty}(1/n)E\Sigma_{k=1}^{n}Q(t_k)$ *over all selection sequences (where* $\{t_k, k \geqslant 1\}$ *are the external arrival epochs before selection/rejection).*

Proof. See Hajek (1985). □

Observe that if the arrival process is Poisson then the cost function in theorem 4 is the same as the time average queue length, and hence by Little's Law $\{r_k^*\}$ also minimises mean delay. For non-Poisson arrivals, the cost function is the average number of customers seen by all arrivals (whether they are accepted or not), and thus minimising it does not, in general, minimise the mean delay of accepted customers. The import of this cost function in these cases is not clear.

2.4 *Static policies*

In static policies, no information is used about system state or past allocations. Probabilistic policies that base the allocation probabilities on a priori distributional information are static policies. The most well-studied problem in this class is the probabilistic allocation of tasks arriving in a Poisson stream and requiring general i.i.d. service. The servers have possibly unequal service rates. See Chow & Kohler (1979), de Souza e Silva & Gerla (1984), Tantawi & Towsley (1984), Ni & Hwang (1985) and Bonomi & Kumar (1990). In each case the cost function is the mean task sojourn time over all task streams. Since probabilistic splitting retains the Poisson nature of the arrivals, the mean task sojourn time can be written down explicitly. Minimisation of this over the splitting probabilities is a nonlinear programming problem that can be solved with the Lagrange multiplier technique. Explicit results can be obtained and are given, for example, in Bonomi & Kumar (1990).

In addition to obtaining the explicit solution of the mean delay minimisation problem, it is also shown in Bonomi & Kumar (1990) that the minimisation problem is equivalent to least squares idle time balancing of the queues. This result is then used to develop an adaptive algorithm for on-line learning of the optimal splitting probabilities, via a stochastic approximation technique.

3. Multitask jobs with precedence constraints

The basic system structure is still that shown in figure 2, except that now each job comprises a batch of tasks with precedence constraints between them. Figure 4 shows a batch of 7 tasks $\{T_1, T_2, \ldots, T_7\}$ with precedence constraints between them; for example, the directed edges between T_3 and T_6 and T_4 and T_6 indicate that processing of T_6 cannot begin until T_3 and T_4 are fully processed. This relation is symbolically denoted by $T_3 \succcurlyeq T_6$ and $T_4 \succcurlyeq T_6$. Generically a precedence relation is denoted by Γ. It is assumed that the graph representation of this partial order (as in figure 4) yields a *directed acyclic graph.*

A job is complete only when all its constituent tasks have been served. Thus job sojourn time is the time between the arrival of the job and the completion of processing of the last task in the job to be served.

Essentially all the work done on this problem has been for the following subclass of models. The job interarrival times are i.i.d., and the ith job brings a random batch

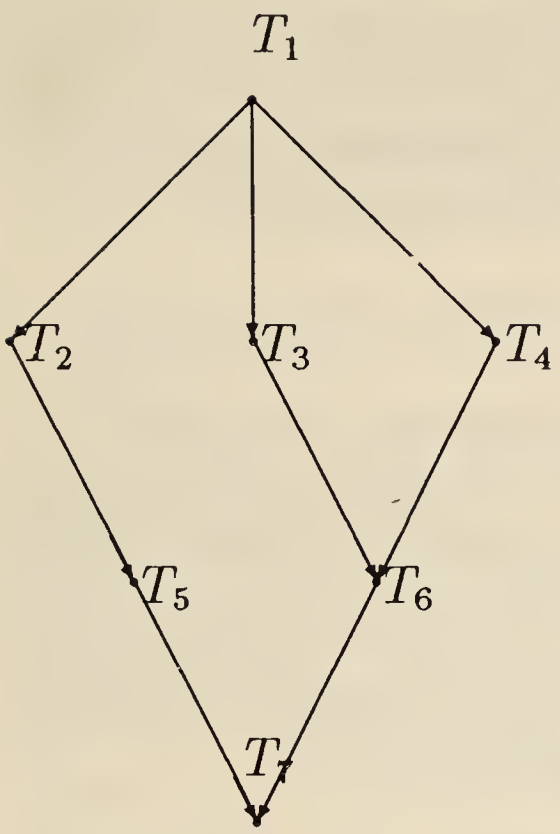


Figure 4. A directed acyclic graph representation of precedence constraints between tasks in a job.

of tasks of size C_k, where the $\{C_k, k \geqslant 1\}$ form an i.i.d. sequence. The "precedence" constraints are simple; all the tasks can be processed independently without having to wait for each other, but the job finishes processing only after all its tasks have been fully served. This is the class of *fork-join* models. Most of the results in the literature are for the special case in which $C_k = N$ (the number of servers) for all $k \geqslant 1$, and upon arrival, one task is assigned to each of the N queues. See Flatto & Hahn (1984), Nelson & Tantawi (1988) and Baccelli *et al* (1985, 1989). These papers essentially concern themselves with exact or approximate analysis of the fork-join queue. Recently some of these results have been extended to nondeterministic task batch sizes (see Shorey 1990). The latter work also studies various task allocation policies, and the effect of task batch size distributions.

In this survey we shall consider the special case of renewal job arrival epochs, i.i.d. task batch sizes, i.i.d. task service times, and servers with identical service rates. An exact analysis of this problem is extremely difficult even for the simplest case of two queues, Poisson arrivals, and exponential service times (Flatto & Hahn 1990). Subsequently, researchers have limited themselves to finding bounds and approximations.

As before the performance measure of interest is job sojourn time. The following stability condition assures the existence of a limiting distribution for the sojourn time.

Let B denote the task service time random variable, and λ the job arrival rate.

Theorem 5. *There exists a task service discipline such that the fork-join system is stable, if* $\lambda EC\, EB < N$.

Proof. The result (Shorey 1990), in fact, holds for the general system with arbitrary precedence relation Γ_k for the batch C_k. Since Γ_k is a directed acyclic graph, there exists a $\bar{\Gamma}_k$ that is a directed chain, and each precedence relation in Γ_k is contained in $\bar{\Gamma}_k$. If we serve the tasks in C_k as if they had the precedence relation $\bar{\Gamma}_k$, then we just have a GI/GI/N queue with traffic intensity $\lambda ECEB$. This queue is stable under the stated condition. Hence, we have a task service discipline that under the given condition renders the system stable. □

First consider the case with $C_k = N, \forall k \geqslant 1$, and one arriving task is allocated to each queue. Assuming that the stability condition holds, each queue is now a stable GI/GI/1 queue. Denote by $W^{(i)}(t)$ the stationary waiting time random variable at the ith queue, $1 \leqslant i \leqslant N$. Then the stationary job sojourn time T is given by

$$T = \max_{1 \leqslant i \leqslant N} (W^{(i)} + B^{(i)}),$$

where $B^{(i)}$, $1 \leqslant i \leqslant N$ are i.i.d., with the distribution of B, the task service time.

The distributions of $W^{(i)}$, $1 \leqslant i \leqslant N$ can be determined explicity for many special cases, but the difficulty is that $\{W^{(i)}, 1 \leqslant i \leqslant N\}$ are not mutually independent. Bounds, however, can be developed as follows. First replace the arrival process of jobs with a process with deterministic interarrival times equal to $1/\lambda$ (where λ is the arrival rate). Let $\mathbf{W}^{(i)}, 1 \leqslant i \leqslant N\}$ denote the steady-state waiting time random variables for the new system. It is clear that $\{\mathbf{W}^{(i)}, 1 \leqslant i \leqslant N\}$ are mutually independent. Let

$$\mathbf{T} = \max_{1 \leqslant i \leqslant N} (\mathbf{W}^{(i)} + B^{(i)}).$$

Further, let $\{\bar{W}^{(i)}, 1 \leqslant i \leqslant N\}$ be mutually independent with $\bar{W}^{(i)}$ and $W^{(i)}$ having the same distribution for each i. Let

$$\bar{T} = \max_{1 \leqslant i \leqslant N} (\bar{W}^{(i)} + B^{(i)}).$$

Theorem 6. *$\mathbf{T} \leqslant_c T \leqslant_{st} \bar{T}$, where $\leqslant_c$ denotes the convex increasing ordering.*

Proof. The lower bound intuitively (Baccelli *et al* 1989) states that "determinism minimises queueing delay". The upper bound uses the fact (proven in Baccelli *et al* 1989) that $(W^{(i)}, 1 \leqslant i \leqslant N)$ are associated (see Barlow & Proschan 1975). □

The significance of this theorem is that it yields computable bounds to the mean of T, since on each side we have the maximum of independent random variables.

The upper bound has been extended in Shorey (1990) to random batch size C, and probabilistic allocation of tasks to queues (i.e., there are probabilities $p_i, 1 \leqslant i \leqslant N$, $0 \leqslant p_i \leqslant 1$, $\Sigma_{1=i}^{N} p_i = 1$, such that an arriving batch is multinomially partitioned over the N queues using these probabilities). Let $\{C^{(i)}, 1 \leqslant i \leqslant N\}$ be the subbatches allocated to the queues. The main difficulty in extending theorem 6 is that the subbatches are dependent, in general. Note that if C is Poisson then the subbatches are independent and theorem 6 applies. Let $W^{(i)}$ denote the stationary waiting time at the ith queue, let $D^{(i)}$ denote the total service time of the subbatch assigned to the ith queue, and let $C^{(i)}$ denote the subbatch size assigned to the ith queue. Then the job sojourn time is given by

$$T = \max_{1 \leqslant i \leqslant N} (W^{(i)} + D^{(i)}) I_{\{C^{(i)} > 0\}},$$

where $I_{\{\cdot\}}$ is the indicator of $\{.\}$. In the expression for T, each term inside the maximum is multiplied by $I_{\{C^{(i)}>0\}}$ since a possibly empty batch may be assigned to a queue. Let $\{X^{(i)}, 1 \leqslant i \leqslant N\}$ be mutually independent, such that $X^{(i)}$ has the same distribution as $(W^{(i)} + D^{(i)}) I_{\{C^{(i)}>0\}}$. Let $\bar{T} = \max_{1 \leqslant i \leqslant N} X^{(i)}$.

Theorem 7. *If C has a geometric distribution then $T \leqslant_{st} \bar{T}$.*

Proof. The proof (Shorey 1990) rests on the result that the components of a multinomial partition of a geometric random variable are associated. □

Numerical results presented in Shorey (1990) show that the upper bound to ET yielded by theorem 7 is also a good approximation to ET. It is found that for the same mean batch size, a geometric batch size distribution yields larger mean delay than a Poisson batch size. This may be attributed to the larger variability in the geometric distribution.

Some simple, albeit loose lower bounds for the mean job delay in the general fork-join delay model are also given in Shorey (1990).

Simulations have been used to compare three task allocation strategies: probabilistic (as discussed above), cyclic (tasks are assigned to queues in a round-robin fashion), and JSQ (tasks arriving in a batch are successively assigned to the shortest queue). In each example studied it is found that the JSQ yields the least mean job delay, and the probabilistic strategy, the most. Thus the ordering observed in § 2 seems to be preserved, but we do not have a proof for this, nor is it known whether JSQ is still optimal in any sense.

The simulations yielded an interesting observation. When JSQ as just described (JSQ by task) was compared with JSQ by batch (i.e., the entire arriving batch of tasks is put into the shortest queue), it was found that whereas at light and moderate loads JSQ by task performed better, at heavy loads the performance of the two became equal. This has important implications since, in practice, JSQ by task requires interprocessor communication overheads, which we do not incorporate in our analysis. Thus at heavy loads it may be better to switch the allocation policy to JSQ by batch. This simulation result may be explained in terms of the heavy traffic results for JSQ discussed in § 2·2.

4. Stream jobs

A stream job comprises a sequence of tasks that arrive over time. Each task can be served independently at any of the N servers, and hence upon arrival it may be allocated to any of the N queues. The constraint is that the tasks must depart the system in sequence. Consequently, there is a need for a *resequencing buffer* following the servers (see figure 5). A stream job models packet arrivals in a virtual circuit in a computer network.

Very few results are available on this problem. Most of the available results concern a single stream job extending over infinite time, i.e., a stream of arrivals that must be served by any of the servers and must depart in sequence. Such models have been studied in Varma (1987), Jean-Marie (1988), Gün & Jean-Marie (1990) and Jean-Marie & Gün (1990). Recently some results have been obtained for the more realistic problem of a sequence of stream jobs each of finite extent; it is only necessary to maintain order between tasks belonging to a particular job (Kuri 1990). Further, as in § 3, most of the work concerns itself with the analysis of task delays under various assumptions and allocation policies. Some results regarding "good" allocation policies are discussed in Kuri & Kumar (1991).

Jean-Marie (1988) considers a problem with two independent Poisson streams of

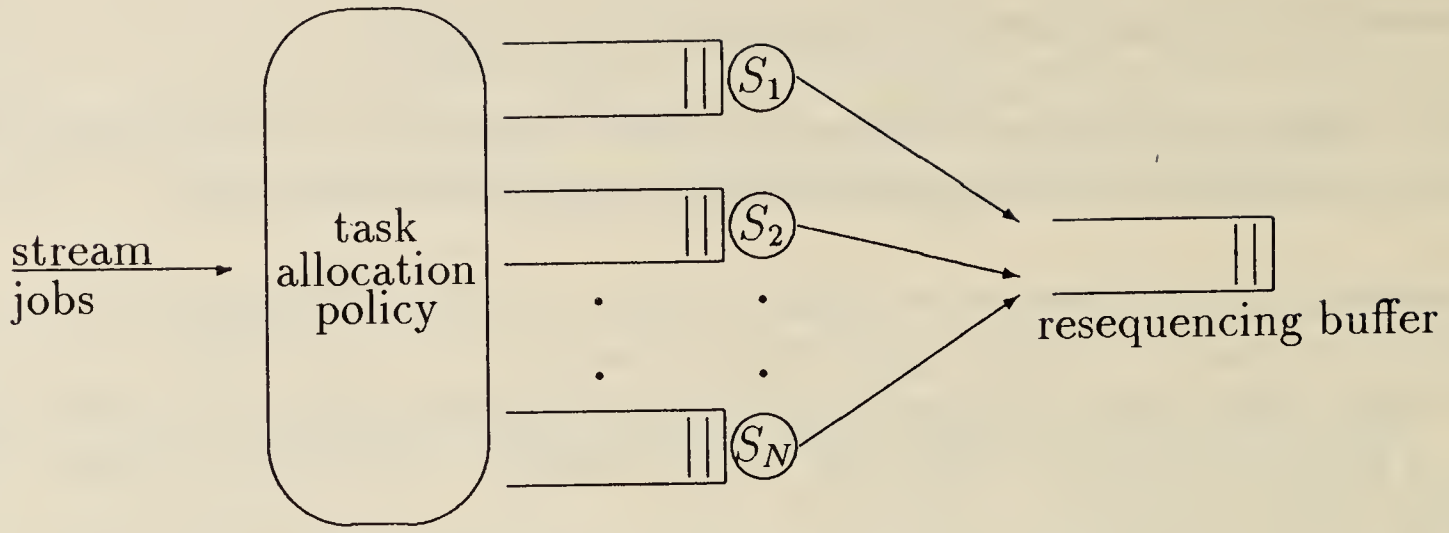


Figure 5. Stream jobs require a resequencing buffer.

tasks, that arrive to two queues with equal speed servers. The tasks in both the streams have i.i.d. service requirements with an exponential distribution. A task from stream 1 is allocated to queue 1 with probability p_1, and a task from stream 2 is allocated to queue 1 with probability p_2. The mean sojourn time, over all tasks of either stream, is derived as a function of p_1 and p_2. The problem of finding (p_1, p_2) that minimises this mean sojourn time is then studied.

The as yet unpublished results in Varma (1987), Gün & Jean-Marie (1990) and Jean-Marie (1990) are reported on in a survey article (Baccelli & Makowski 1990). These studies provide stochastic comparison results that help in bounding the delay performance of a single task stream that must be allocated to N queues, and must depart in order. For example, similar in spirit to the result in §2·1, it is shown that the delay through N G/G/1 queues is stochastically lower bounded by a system in which there are an infinite number of servers (the G/G/∞ system).

Motivated by performance comparison between virtual circuit and datagram packet networks (Tanenbaum 1988), some simple models are studied in Kuri (1990). A job (or call) has a holding time that is exponentially distributed with parameter μ; during a call, tasks (or packets) arrive at the rate λ. Each packet can be put into one of two queues with identical service rates; the packets have an exponential service time distribution with mean 1. Packets within a call must emerge from the queueing system in sequence. After the completion of a call another call arrives immediately. It is easy to see that the aggregate packet arrival process is Poisson with rate λ.

Several strategies are studied. The entire call may be allocated to one queue (this is the virtual circuit case), or packets from calls may be assigned to either queue upon arrival (the datagram case). For each case three allocation strategies are studied: probabilistic, alternating and join-the-shortest packet-queue. It is found in every case that, in spite of resequencing delays, datagram networks perform better in the sense of lower mean packet delay. Further JSQ yielded the smallest mean packet delay in both the virtual circuit and datagram cases.

Thus purely from a queueing delay point of view, resequencing delay in the case of stream jobs do not outweigh the gain due to load balancing obtained by splitting the job over several queues.

The question then arises as to the best allocation strategy for tasks in a stream job. Some progress has been made on this problem and is reported in Kuri & Kumar (1991). A single stream job of infinite extent is considered, i.e., tasks arrive in a stream and must depart in sequence. There are N equal-speed servers, each with its own queue. Each task must be allocated to a queue upon arrival, and is served to completion at that queue, in an FCFS manner. For a general arrival process, the special case of *deterministic service times* is considered (i.e. all tasks have the same fixed service time

at any server). Let d'_n denote the departure epoch of the nth arriving task with an arbitrary task allocation policy, and d_n denote the departure epoch of the nth arriving task with the join-the-shortest-work-queue (JSWQ) policy (i.e., an arriving task joins the queue with the least work). Note that d'_n and d_n include resequencing delay. We assume that all servers have zero work at time 0.

Theorem 8. *On every sample path,* $\forall n, d'_n \geqslant d_n$.

Proof. Denoting by c'_n, c_n the service departure epochs of the nth arriving task, by γ'_n, γ_n the nth service departure epochs, by γ''_n the nth service departure epoch in the G/D/N system,

$$d'_n = \max_{1 \leqslant i \leqslant N} c'_n \geqslant \gamma'_n \geqslant \gamma''_n = \gamma_n = c_n = d_n,$$

where the first relation is by definition, the second is obvious, the third follows from Wolff (1977) and the remaining from results in Kuri & Kumar (1991). □

Let $\tilde{d}_n$ denote the departure epoch of the nth arriving task, in the system in which tasks are allocated cyclically, or in a round-robin fashion, to the N queues.

Theorem 9. *On every sample path,* $\forall n, \tilde{d}_n = d_n$, *i.e., departures with cyclic allocation occur at the same epochs as with* JSWQ.

Proof. See Kuri & Kumar (1991). □

The last theorem asserts the optimality of the cyclic rule for deterministic service times. We note that JSQ may be strictly worse in the sense of sample-path-wise comparison.

Our discussion and results thus imply the following more general fact. Suppose tasks are arriving in several streams of finite or infinite extent; the tasks in a stream may all arrive instantaneously (batch arrivals), or in smaller batches, or spread over time. Only the tasks within a stream need to depart in sequence. If the service times of all the tasks are deterministic and identical, then the policy of mixing all the streams into one stream, and allocating tasks in this stream in a cyclic fashion, is sample-path-wise optimal.

5. Final remarks

It is clear from the foregoing that there are several interesting problems in this class of models that deserve further study. The JSQ result is quite general, but, as a practical matter, the scheduler that allocates tasks to queues might not be able to observe the instantaneous queue lengths. It will be interesting to understand the effect of delayed information on the optimality of JSQ. Hajek's (1985) result on the splitting of a renewal process by regular sequences needs to be extended to obtain policies for sojourn time optimal semidynamic routing to multiple queues.

The analysis of task allocation strategies for multitask jobs is still a topic of current research, particularly when the jobs do not have the simple deterministic fork-join

structure with one task being allocated to each queue. The problem of good allocation policies in the multitask case has received very little attention.

References

Agrawala A, Tripathi S 1981 On the optimality of semidynamic routing schemes. *Inf. Process. Lett.* 13: 20–22

Baccelli F, Makowski A M 1990 Synchronization in queuing systems: In *stochastic analysis of computer and communication system* (ed.) H Takagi (Amsterdam: North Holland/Elsevier Science Publishers)

Baccelli F, Makowski A M, Schwartz A 1985 Simple computable bounds and approximations for the fork-join queue. *Proc. Int. Seminar on Computer Networking and Performance Evaluation*

Baccelli F, Massey W A, Towsley D 1989 Acyclic fork-join queuing networks. *J. Assoc. Comput. Mach.* 36: 615–642

Barlow R E, Proschan F 1975 *Statistical theory of reliability and life testing: Probability models* (New York: Holt, Rinehart and Winston)

Bonomi F, Kumar A 1990 Adaptive optimal load balancing in a non-homogeneous multiserver system with a central job scheduler. *IEEE Trans. Comput.* 39: 1232–1250

Chow Y C, Kohler W 1979 Models of dynamic load balancing in a heterogeneous multiple processor system. *IEEE Trans. Comput.* 28: 354–361

de Souza e Silva E, Gerla M 1984 Load balancing in distributed systems, with multiple classes and site constraints. *Performance 84* 17–33

Eager D L, Lazowska E D, Zahorjan J 1986 Adaptive load sharing in homogeneous distributed systems *IEEE Trans. Software Eng.* 12: 662–675

Ephremides A, Varaiya P, Walrand J 1980 A simple dynamic routing problem: *IEEE Trans. Autom. Control.* 25: 690–693

Flatto L, Hahn S 1984 Two parallel queues created by arrivals with two demands I: *SIAM J. Appl. Math.* 44: 1041–1053

Flatto L, McKean H P 1977 Two queues in parallel. *Commun. Pure Appl. Math.* 30: 255–263

Flores C 1990 Diffusion approximations for computer communications networks: In *Stochastic analysis of computer and communication systems* (ed.) H Takagi (Amsterdam: North Holland/Elsevier Science)

Foschini G J 1977 On heavy traffic diffusion analysis and dynamic routing in packet switched networks. In *Computer performance* (eds) K M Chandy, M Reiser (Amsterdam: North Holland)

Foschini G J, Salz J 1978 A basic dynamic routing problem and diffusion. *IEEE Trans. Commun.* 26: 320–327

Gün L, Jean-Marie A 1990 Resequencing in parallel M/G/1 queues with Bernoulli loading. *Oper. Res.* (submitted)

Hajek B 1984 Optimal control of two interacting service stations. *IEEE Trans. Autom. Control.* 29: 491–499

Hajek B 1985 Extremal splitting of point processes. *Math. Oper. Res.* 10: 543–556

Jean-Marie A 1988 Load balancing in a system of two queues with resequencing. *Performance '87* 75–88

Jean-Marie A, Gün L 1990 Asymptotic results for parallel queues with resequencing. *J. Assoc. Comput. Mach.* (submitted)

Kemae T, Krengel U, O'Brien G L 1977 Stochastic inequalities on partially ordered spaces. *Ann. Probab.* 5: 899–912

Kuri J 1990 *Comparative performance evaluation of routing strategies in connection-oriented and connectionless data networks.* M E thesis, Electrical Communication Engineering Department, Indian Institute of Science, Bangalore

Kuri J, Kumar A 1991 On the optimal allocation of customers that must depart in sequence. *Oper. Res. Lett.* (submitted)

Lin W, Kumar P R 1987 Optimal control of a queuing system with two heterogeneous servers. *IEEE Trans. Autom. Control.* 29: 696–703

Mirchandaney R, Towsley D, Stankovic J A 1989 Analysis of the effects of delays on load sharing. *IEEE Trans. Comput.* 38: 1513–1525

Nelson R D, Philips T K 1990 An approximation for the mean response time for shortest queue routing with general interarrival and service times. IBM Research Report, RC-15429 (#68659)

Nelson R, Tantawi A N 1988 Approximate analysis of fork join synchronisation in parallel queues. *IEEE Trans. Comput.* 37: 739–743

Ni L M, Hwang K 1985 Optimal load balancing in a multiple processor system with multiple job classes. *IEEE Trans. Software Eng.* 11: 491–496

Reiman M I 1983 Some diffusion approximation with state space collapse. *Modelling and performance evaluation methodology* (eds) F Baccelli, G Fayolle (Lecture Notes in Control and Inf. Science) Vol. 60. (Berlin: Springer Verlag)

Ross S 1983 *Stochastic processes* (New York: John Wiley)

Shorey R 1990 *Performance analysis and scheduling of stochastic fork-join jobs in a multicomputer system.* M Sc (Eng). thesis, Electrical Communication Engineering Department, Indian Institute of Science, Bangalore

Smith D R, Whitt W 1981 On the efficiency of shared resources in queuing systems. *Bell Syst. Tech. J.* 60: 39–56

Stoyan D 1983 *Comparison methods for queues and other stochastic models* (New York: John Wiley)

Tanenbaum A 1988 *Computer networks*, 2nd edn (Englewood Cliffs, NJ: Prentice Hall)

Tantawi A N, Towsley D 1984 Optimal load balancing in distributed computer systems. Tech. Rep. RC-10346, IBM T J Watson Research Center

Varma S 1987 *Some problems in queuing systems with resequencing*, M S thesis, Electrical Engineering Dept., Univ. of Maryland, College Park, Maryland; Also Tech. Rep. TR-87-192, Systems Res. Center, Univ. of Maryland, College Park.

Walrand J 1988 *An introduction to queuing networks* (Englewood Cliffs, NJ: Prentice Hall)

Weber R R 1978 On the optimal assignment of customers to parallel queues. *J. Appl. Probab.* 15: 406–413

Whitt W 1986 Deciding which queue to join: Some counterexamples. *Oper. Res.* 34: 55–62

Winston W 1977 Optimality of the shortest line discipline. *J. Appl. Probab.* 14: 181–189

Wolff R W 1977 An upper bound for multi-channel queues. *J. Appl. Probab.* 14: 884–888

Wolff R W 1989 *Stochastic modelling and the theory of queues* (Englewood Cliffs, NJ: Prentice Hall)

Yum T S 1981 The design and analysis of a semidynamic deterministic routing rule: *IEEE Trans. Commun.* 29: 498–504

Yum T S, Schwartz M 1981 The join-biased queue rule and its application to routing in computer communication networks. *IEEE Trans. Commun.* 29: 505–511

Recent results on least squares-based adaptive control of linear stochastic systems in white noise

WEI REN and P R KUMAR

Coordinated Science Laboratory, University of Illinois, 1101, West Springfield Avenue, Urbana, Illinois 61801, USA

Abstract. Recently, progress has been made on establishing the stability and performance of linear stochastic systems when they are adaptively controlled in a certainty equivalent fashion using least squares- or extended least squares-based parameter estimates. Here we provide an overview of these results.

We consider first the case of white gaussian noise, where the convergence of the parameter estimates can be established for generically all systems. Then we provide an account of the stability and performance of certainty equivalent controllers for which parameter convergence has been established.

Next we turn to the white non-gaussian case, and obtain upper bounds for the parameter error and the normalized prediction error. Finally we exploit these bounds for the self-tuning regulator when "b_0" is known and the delay equals one.

Keywords. Adaptive control; system identification; recursive parameter estimation; least-squares estimates; stochastic adaptive control.

1. Introduction

Adaptive control deals with the problem of controlling an unknown system. As one acquires data concerning inputs and outputs in real-time, one acquires information about the initially unknown system, which can be used to refine the controller being used. A popular procedure for adaptive control consists of making real-time estimates of the unknown parameters describing the system, and then at each time step using these estimated parameters to determine an appropriate control input.

Since the seminal work of Åström & Wittenmark (1973), there has been a great deal of interest in establishing the stability and convergence of adaptive controllers for linear stochastic systems, which use least squares-based parameter estimation schemes.

While much progress has been made for adaptive controllers using stochastic gradient parameter estimation schemes, see Goodwin *et al* (1981) where the optimality of an adaptive minimum variance controller was established, and Becker *et al* (1985) where parameter convergence and self-tuning were proved, such results have been elusive for the corresponding least squares-based schemes, until recently.

Early on, Sternby (1977) and Rootzen & Sternby (1984) had introduced the idea

of Bayesian embedding for the *white* gaussian noise case, which allows one to establish the convergence of the parameter estimates. Recently, these results have been exploited to establish the stability and quantify the performance of some certainty equivalent adaptive controllers, see Kumar (1990). More recently, Guo & Chen (1990) have succeeded in establishing the optimality of some adaptive control schemes, including a version of the self-tuning redulator. Their results exploit the properties of the least squares estimator established by Lai & Wei (1982) and the extended least squares-estimator studied by Solo (1979), Chen & Guo (1986) and Lai & Wei (1986).

In this paper we provide an account of these contributions.

2. Bayesian embedding

Let us consider the system:

$$y_{t+1} = \phi_t^T \theta^0 + w_{t+1},$$

where $\{w_t\}$ is a sequence of independent, identically distributed random variables, each $N(0, \sigma^2)$, i.e., a standard white gaussian noise sequence, and ϕ_t is $\sigma(y_0, \ldots, y_t)$-measurable. (The strong restriction of gaussianity should be noted.)

The recursive least squares parameter estimator for θ^0 is given by,

$$\hat{\theta}_{t+1} = \hat{\theta}_t + R_t^{-1} \phi_t [y_{t+1} - \phi_t^T \hat{\theta}_t]; \quad \hat{\theta}(0) = \bar{\theta} \tag{1}$$

$$R_{t+1} = R_t + \phi_{t+1} \phi_{t+1}^T; \quad R^{-1}(-1) > 0. \tag{2}$$

The analysis of these parameter estimates is generally quite difficult, but Sternby (1977) and Rootzen & Sternby (1984) have introduced the following Bayesian embedding procedure which quickly allows us to obtain a considerable amount of information rather easily.

Let us presume that θ^0 is itself a gaussian random variable, $N(\bar{\theta}, R_{-1}^{-1})$, which is independent of w. Then the recursive least squares algorithm (1), (2) coincides with the Kalman filter algorithm for estimating θ^0 from observations of $\{y_i, \phi_{i-1}\}$. Since the Kalman filter for this system computes the conditional mean of θ^0 [see Chen *et al* (1989) where this is rigorously established without requiring $E \| \phi_t \|^2 < + \infty$], it follows that

$$\hat{\theta}_t = E[\theta^0 | y_0, \ldots, y_t].$$

This, however, shows that $\{\hat{\theta}_t\}$ is a martingale sequence which converges with probability one. Since, however, the probability space has been extended by introducing randomness into the choice of θ^0, what we can conclude is only applicable to all θ^0's except for a zero Lebesgue measure set.

Theorem. (*Sternby*) *The recursive least squares parameter estimates converge a.s., except possibly when θ^0 is in a null Lebesgue measure set.*

3. The consequences for minimum phase systems when parameter estimates converge

Let us now consider the consequences for adaptive control when the parameter estimates are convergent. As we have seen above, this is generically the case whenever the noise is gaussian as well as white.

Suppose therefore that $\{w_t\}$ is a white noise sequence satisfying

(i) $E(w_t|w_0,\ldots,w_{t-1})=0$

(ii) $E(w_t^2|w_0,\ldots,w_{t-1})=\sigma^2$ a.s.

(iii) $\sup E(|w_t|^\alpha|w_0,\ldots,w_{t-1})<+\infty$ a.s. for some $\alpha>2$.

Note that we do *not* assume that w_t is gaussian. Instead we will directly suppose in this section that

$$\hat{\theta}_t\to\theta^* \text{ a.s.} \tag{3}$$

where θ^* is a finite random variable.

We start by recalling that the recursive least squares parameter estimates produced by (1), (2) can be rewritten as

$$\left(\sum_0^{N-1}\phi_t\phi_t^T\right)\hat{\theta}_N=\left(\sum_0^{N-1}\phi_t y_{t+1}\right)+R(-1)(\bar{\theta}-\hat{\theta}_N).$$

Let

$$r_t:=\operatorname{trace}R_t.$$

By utilizing the convergence property (3) it follows as in Kumar (1990) that

$$(1/r_{N-1})\sum_0^{N-1}\phi_t\phi_t^T\tilde{\theta}_t\to 0 \quad \text{a.s.} \tag{4}$$

as well as

$$(1/r_{N-1})\left(\sum_0^{N-1}\phi_t\phi_t^T\right)\tilde{\theta}^*\to 0 \quad \text{a.s.} \tag{5}$$

where $\tilde{\theta}_t:=\hat{\theta}_t-\theta^0, \tilde{\theta}^*:=\theta^*-\theta^0$.

Let us now examine the consequences of (4), (5) and (3) for adaptive control.

Thus suppose that we have a linear stochastic system,

$$A(q^{-1})y_t=q^{-d}B(q^{-1})u_t+w_t, \tag{5'}$$

where $A(q^{-1})=1-a_0q^{-1}-\cdots-a_pq^{-1}$ and $B(q^{-1})=b_0+b_1q^{-1}+\cdots+b_sq^{-s}$ are polynomials in the backward shift operator q^{-1}. Using the obvious notation, let us denote by $\hat{A}_t(q^{-1})$ and $\hat{B}_t(q^{-1})$ the polynomials formed from the estimated coefficients in $\hat{\theta}_t$, and by $A^*(q^{-1})$ and $B^*(q^{-1})$ the coresponding polynomials from θ^*. Then (4) can be written as

$$(1/r_{N-1})\sum_0^{N-1}[\hat{A}_t(q^{-1})y_{t+1}-\hat{B}_t(q^{-1})q^{-d}u_{t+1}-w_{t+1}]^2\to 0 \quad \text{a.s.,} \tag{6}$$

while (5) gives,

$$(1/r_{N-1})\sum_0^{N-1}[A^*(q^{-1})y_{t+1}-q^{-d}B^*(q^{-1})u_{t+1}-w_{t+1}]^2\to 0 \quad \text{a.s.} \tag{7}$$

Let us now suppose that the true system is of strictly minimum phase, i.e., $B(q^{-1})$ has all roots inside the open unit disk. Then it follows that

$$r_{N-1}=0\left(\sum_0^N y_t^2\right),$$

and so (6) and (7) can be sharpened by replacing r_{N-1} by $\Sigma_0^N y_t^2$.

4. Stability and performance of certainty equivalence adaptive controllers when parameter estimates converge

Let us now consider the type of adaptive control law used. Suppose that we use a control law of the form

$$\hat{R}(q^{-1})\hat{B}(q^{-1})u_t = \hat{S}(q^{-1})y_t + \hat{T}(q^{-1})z_t, \tag{8}$$

where $\{z_t\}$ is a deterministic, bounded reference signal, and $\hat{R}$, $\hat{S}$ and $\hat{T}$ are obtained by some design rule or mapping from the estimated model $\hat{A}$ and $\hat{B}$, and the mapping satisfies the following conditions:

(i) it is continuous;
(ii) $\hat{H}(q^{-1}) := \hat{A}(q^{-1})\hat{R}(q^{-1}) - q^{-d}\hat{S}(q^{-1})$ has all its roots inside the open unit disk.

The key restriction of the control law (8) lies in the fact that $\hat{B}(q^{-1})$ is a factor of the polynomial acting on u_t. It corresponds to a nominal zero cancellation scheme. This restriction is employed here merely to avoid the singularity problem when $\hat{A}_t$ and $\hat{B}_t$ have a common factor. The above certainty equivalent control law includes the self-tuning regulator for the white noise case, pole-zero placement schemes etc.

Multiplying through the summation in (6) by $\hat{R}_t$ and making use of the fact that $\hat{H}_t$, $\hat{T}_t$ and $\hat{R}_t$ are convergent, it is easy to establish that

$$\left\{\sum_0^{N-1} [H^*(q^{-1})y_{t+1} - q^{-d}T^*(q^{-1})z_{t+1} - R^*(q^{-1})w_{t+1}]^2\right\} \div$$
$$\div\left(\sum_0^{N-1} y_{t+1}^2\right) \to 0 \quad \text{a.s.} \tag{9}$$

where H^*, T^* and R^* are the limits of $\hat{H}_t$, $\hat{T}_t$ and $\hat{R}$ respectively, corresponding to θ^*.

From this, by using the stability of $1/[H^*(q^{-1})]$, it is an easy matter to establish that

$$\limsup (1/N) \sum_0^{N-1} (y_t^2 + u_t^2) < +\infty \quad \text{a.s.},$$

i.e., that the system is "stable" (in the sense of mean square boundedness).

Moreover, since $\Sigma_0^{N-1} y_{t+1}^2 = O(N)$, we can refine (9) to,

$$\lim (1/N) \sum_{t=0}^{N} [H^*(q^{-1})y_t - q^{-d}T^*(q^{-1})z_t - R^*(q^{-1})w_t]^2 = 0 \quad \text{a.s.}$$

This result provides "self-optimality" conclusions. (For example, when $d = 1$, let $\hat{H}(q^{-1}) \equiv 1$, $\hat{T}(q^{-1}) \equiv 0$, $\hat{R}(q^{-1}) \equiv 1$, as in the case of the self-tuning regulator, then one has $(1/N)\Sigma_0^N (y_t - w_t)^2 \to 0$, which proves the "minimum variance" property of the output.)

Moreover, we can also obtain "self-tuning" results. From (7), we have $(1/N)\Sigma_0^{N-1}[\tilde{A}^*(q^{-1})y_{t+1} - \tilde{B}^*(q^{-1})q^{-d}u_{t+1}]^2 \to 0$, where $\tilde{A}^* := A^* - A$. Now note that whenever one has $(1/N)\Sigma_0^{N-1}(\alpha_i w_{i+1} + \zeta_i)^2 \to 0$, where α_i and ζ_i are $\sigma(w_0, \ldots, w_i)$-measurable, then one must have $(1/N)\Sigma_0^{N-1}\alpha_i^2 \to 0$. Exploiting this, it is straightforward to show the following additional results:

$$[A^*(q^{-1}) - A(q^{-1})]R^*(q^{-1})B^*(q^{-1})$$
$$= q^{-d}[B^*(q^{-1}) - B(q^{-1})]S^*(q^{-1}), \quad \text{a.s.}$$
$$a_i^* = a_i \text{ for } 0 \leqslant i \leqslant d-2 \quad \text{a.s.},$$

and

$$\lim (1/N) \sum_{0}^{N-1} [(B^*(q^{-1}) - B(q^{-1})\, T^*(q^{-1}))z_t]^2 = 0 \quad \text{a.s.}$$

This last result allows us to exploit any known "persistency of excitation" properties about $\{z_t\}$.

For precise details on the above proofs as well as examples to illustrate how they are used, we refer the reader to Kumar (1990).

5. Identification in white noise

We have seen above that if $\{w_t\}$ is gaussian and white, then we can (generically) show that the least-squares parameter estimates converge. Now we will consider the behaviour of the least squares parameter estimates when the gaussianity assumption is removed. The results below are drawn from Lai & Wei (1982), to which the reader is referred for further details, results and applications.

Let us consider the function,

$$V_t := \tilde{\theta}_t^T R_{t-1} \tilde{\theta}_t.$$

It is trivial to observe that

$$\|\tilde{\theta}_t\|^2 = O\{V_t/[\lambda_{\min}(R_{t-1})]\},$$

where $\lambda_{\min}(R)$ denotes the minimum eigenvalues of R. Let us now calculate an upper bound for the rate of growth of V_t.

Note that $\tilde{\theta}_t$ can be written as,

$$\tilde{\theta}_t = R_{t-1}^{-1}\left[\sum_{i=0}^{t-1} \phi_i w_{i+1} + R(-1)(\bar{\theta} - \theta^0)\right].$$

Simple algebra then shows that,

$$V_{N+1} = V_N - \frac{|\phi_N^T \tilde{\theta}_N|^2}{1 + \phi_N^T P_{N-1} \phi_N} + 2\left(\frac{\phi_N^T \tilde{\theta}_N}{1 + \phi_N^T P_{N-1} \phi_N}\right) w_{N+1} + \\ + \frac{\phi_N^T P_{N-1} \phi_N}{1 + \phi_N^T P_{N-1} \phi_N} w_{N+1}^2.$$

By telescoping one has,

$$V_{N+1} + \sum_{k=1}^{N} \frac{|\phi_k^T \tilde{\theta}_k|^2}{1 + \phi_k^T P_{k-1} \phi_k} \left\{1 - \left[\sum_{k=1}^{N} \left(\frac{2\phi_k^T \tilde{\theta}_k}{1 + \phi_k^T P_{k-1} \phi_k}\right) w_{k+1}\right] \div \right. \\ \left. \div \left[\sum_{k=1}^{N} \frac{|\phi_k^T \tilde{\theta}_k|^2}{1 + \phi_k^T P_{k-1} \phi_k}\right]\right\}$$

$$= V_0 + \sum_{k=1}^{N} \left(\frac{\phi_k^T P_{k-1} \phi_k}{1 + \phi_k^T P_{k-1} \phi_k}\right) w_{k+1}^2.$$

Now we note that if ζ_k is $\sigma(w_0, \ldots, w_k)$-measurable, then $(\Sigma_{k=1}^{N} \zeta_k w_{k+1})/(\Sigma_{k=1}^{N} \zeta_k^2) \to 0$

a.s., if $\Sigma_{k=1}^{N}\zeta_k^2 \to +\infty$ a.s. Moreover if $\sup_k|\zeta_k| < \infty$, then one also knows that $\Sigma_{k=1}^{N}\zeta_k w_{k+1}^2 = O(\Sigma_{k=1}^{N}|\zeta_k|)$. This allows us to conclude that

$$V_{N+1} = O\left\{\sum_{k=1}^{N}[(\phi_k^T P_{k-1}\phi_k)/(1+\phi_k^T P_{k-1}\phi_k)]\right\},$$

as well as,

$$\left\{\sum_{k=1}^{N}[(|\phi_k^T\tilde{\theta}_k|^2)/(1+\phi_k^T P_{k-1}\phi_k)]\right\}$$

$$= O\left\{\sum_{k=1}^{N}[(\phi_k^T P_{k-1}\phi_k)/(1+\phi_k^T P_{k-1}\phi_k)]\right\}.$$

It therefore remains only to calculate the rate of growth of $\Sigma_{k=1}^{N}[(\phi_k^T P_{k-1}\phi_k)/(1+\phi_k^T P_{k-1}\phi_k)]$. For this, we note that

$$(\det R_{k-1})\det(I + P_{k-1}\phi_k\phi_k^T) = (\det R_k),$$

where $P_k := R_k^{-1}$. From the determinantal relation

$$\det(I + P_{k-1}\phi_k\phi_k^T) = 1 + \phi_k^T P_{k-1}\phi_k,$$

we then obtain

$$\sum_{k=1}^{N}\frac{\phi_k^T P_{k-1}\phi_k}{1+\phi_k^T P_{k-1}\phi_k} = \sum_{k=1}^{N}\frac{\det R_k - \det R_{k-1}}{\det R_k} \leqslant \log\frac{\det R_N}{\det R_0}$$

$$= O[\log\lambda_{\max}(R_N)]$$

$$= O(\log r_N).$$

Thus we obtain the following result.

Theorem. (Lai & Wei 1982)

(i) $$\|\tilde{\theta}_t\|^2 = O\left[\frac{\log\lambda_{\max}(R_{t-1})}{\lambda_{\min}(R_{t-1})}\right],$$

(ii) $$\sum_{k=1}^{N}\frac{|\phi_k^T\tilde{\theta}_k|^2}{1+\phi_k^T P_{k-1}\phi_k} = O\left[\frac{\log\lambda_{\max}(R_N)}{\lambda_{\min}(R_N)}\right].$$

6. Optimality of the self-tuning regulator for known b_0: The white noise case

Ever since 1973, the stability of the self-tuning regulator (STR) of Åström & Wittenmark (1973) has been an open problem which has attracted much research attention. Recently, Guo & Chen (1990) have made a remarkable breakthrough and established the stability and optimality of some versions of the STR. Below we provide a brief outline of their approach and result.

The key and seminal idea of Guo & Chen (1990) is to exploit the property (ii) in the theorem of § 5. Specifically denoting

$$\alpha_k := (|\phi_k^T\tilde{\theta}_k|^2)/(1+\phi_k^T P_{k-1}\phi_k),$$

they demonstrate how one can exploit the fact that

$$\sum_{k=1}^{N}\alpha_k = O(\log r_N).$$

Let us consider the case of (5′) where $d = 1$, b_0 is known, and $B(q^{-1})$ is of strictly minimum phase. Consider the control law

$$b_0 u_k + \phi_k^T \hat{\theta}_k = 0,$$

where because b_0 is known, we use

$$\phi_k := (y_k, \ldots, y_{k-p}, u_{k-1}, \ldots, u_{k-s})^T,$$
$$\theta_0 := (a_0, \ldots, a_p, b_1, \ldots, b_s)^T.$$

Then

$$\begin{aligned} |y_{k+1}|^2 &= |b_0 u_k + \phi_k^T \theta^0 + w_{k+1}|^2 \\ &= |b_0 u_k + \phi_k^T \hat{\theta}_k - \phi_k^T \tilde{\theta}_k + w_{k+1}|^2. \end{aligned}$$

For simplicity of exposition, suppose that the noise w_k is bounded. Then

$$\begin{aligned} |y_{k+1}|^2 &= O(|\phi_k^T \tilde{\theta}_k|^2) + O(1) \\ &= O(\alpha_k(1 + \phi_k^T P_{k-1} \phi_k)) + O(1) \\ &= O(\alpha_k \phi_k^T (P_{k-1} - P_k)\phi_k + \alpha_k \phi_k^T P_k \phi_k + 1) \\ &= O(\alpha_k \lambda_{\max}(P_{k-1} - P_k) \| \phi_k \|^2 + \alpha_k + 1). \end{aligned}$$

By the strict minimum phase property of $B(q^{-1})$, we have $\| \phi_k \|^2 = O(\Sigma_{i=0}^k \lambda^{k-i} y_i^2)$ where $0 \leqslant \lambda < 1$. Let us define $h_k := \Sigma_{i=0}^k \lambda^{k-i} |y_i|^2$. Then

$$h_{k+1} = O[\lambda + \alpha_k \lambda_{\max}(P_{k-1} - P_k)]h_k + O(\log r_k).$$

Exploiting the fact that $\{(P_{k-1} - P_k)\}$ is summable, and $\Sigma_{k-1}^N \alpha_k = O(\log r_N)$, by a solution of the above linear upper bound for h, they show that

$$\| \phi_k \|^2 = O(h_{k+1}) = O(r_k^\varepsilon) \text{ for } 0 < \varepsilon < 1.$$

Now note that

$$\begin{aligned} \sum_0^N |\phi_k \tilde{\theta}_k|^2 &= \sum_0^N \alpha_k (1 + \phi_k^T P_{k-1} \phi_k) \\ &= O(\log r_N) + (\max_{0 \leqslant k \leqslant N} \phi_k^T P_{k-1} \phi_k) O(\log r_N) \\ &= O(\log r_N)[1 + \max_{0 \leqslant k \leqslant N} \| \phi_k \|^2] \\ &= O(\log r_N)[1 + r_N^\varepsilon] \\ &= O[r_N^\varepsilon] \quad \text{(for a slightly larger } \varepsilon). \end{aligned}$$

This allows one to easily conclude the "stability" and optimality, as follows,

$$\begin{aligned} \sum_0^N y_{i+1}^2 &= \sum_0^N (-\phi_i^T \tilde{\theta}_i + w_{i+1})^2 \\ &= O(N) + O\left(\sum_0^N |\phi_i^T \tilde{\theta}_i|^2 \right) \\ &= O(N + r_N^\varepsilon). \end{aligned}$$

By the minimum phase property,

$$r_N = O(r_N^{\varepsilon} + O(N),$$

and so

$$r_N = O(N),$$

allowing us to conclude "stability". Finally

$$\begin{aligned}(1/N)\sum_0^N (y_{i+1} - w_{i+1})^2 &= (1/N)\sum_0^N |\phi_i^T \tilde{\theta}_i|^2 \\ &= O(r_N^{\varepsilon})/N \\ &= O(N^{\varepsilon})/N \to 0,\end{aligned}$$

proving the optimality too!

For extensions as well as more details, we refer the reader to Guo & Chen (1990).

The research reported here has been supported by the US Army Research Office under Contract No. DAAL-03-88-K-0046, by the Joint Services Electronics Program under Contract No. N00014-90-J-1270, and by an International Paper Fellowship for the first author.

References

Åström K J, Wittenmark B 1973 On self-tuning regulators. *Automatica* 9: 185–199

Becker A, Kumar P R, Wei C Z 1985 Adaptive control with the stochastic approximation algorithm: Geometry and convergence. *IEEE Trans. Autom. Control* AC-30: 330–338

Chen H F, Guo L 1986 Convergence rate of least squares identification and adaptive control for stochastic systems. *Int. J. Control* 44: 1459–1476

Chen H F, Kumar P R, van Schuppen J H 1989 On Kalman filtering for conditionally Gaussian systems with random matrices. *Systems Control Lett.* 13: 397–404

Goodwin G C, Ramadge P J, Caines P E 1981 Discrete time stochastic adaptive control. *SIAM J. Control Optimization* 19: 829–853

Guo L, Chen H F 1990 Revisit to Åström-Wittenmark's self-tuning regulator and ELS-based adaptive trackers, Research Report, Academia Sinica, Institute of Systems Sciences

Kumar P R 1990 Convergence of least-squares parameter estimate based adaptive control schemes. *IEEE Trans. Autom. Control* 35: 416–424

Lai T L, Wei C Z 1982 Least squares estimate in stochastic regression with applications to identification and control of dynamic systems. *Ann. Math. Stat.* 10: 154–166

Lai T L, Wei C Z 1986 Extended least squares and their applications to adaptive control and prediction in linear systems. *IEEE Trans. Autom. Control.* AC-31: 898–906

Rootzen H, Sternby J 1984 Consistency in least squares estimation: A Bayesian approach. *Automatica* 20: 471–475

Solo V 1979 The convergence of AML. *IEEE Trans. Autom. Control* AC-24: 958–962

Sternby J 1977 On consistency of the method of least squares using martingale theory. *IEEE Trans. Autom. Control* AC-22: 346–352

Controlled Markov chains with constraints

VIVEK S BORKAR

Department of Electrical Engineering, Indian Institute of Science, Bangalore 560 012, India

Abstract. We consider the ergodic control of a Markov chain on a countable state space with a compact action space in presence of finitely many (say, m) ergodic constraints. Under a condition on the cost functions that penalizes instability, the existence of an optimal stable stationary strategy randomized at a maximum of m states is established using convex analytic arguments.

Keywords. Controlled Markov chains; ergodic control; control under constraints; optimal strategy; stationary strategy.

1. Introduction

Ross (1989) studied ergodic (or 'long-run average cost') control of Markov chains with finite state and action spaces when finitely many (say, m) other ergodic costs are required to satisfy prescribed bounds. Using linear programming arguments, he proved the existence of an optimal strategy randomized at a maximum of m states. [See Hordijk & Kallenberg (1984), Beutler & Ross (1985) and Altman & Shwartz (1990) for related work.] This result is extended in Borkar (1991) to countable state space and compact action space using convex analytic arguments under the hypothesis that the chain be positive recurrent under all stationary randomized strategies and the corresponding invariant probability measures tight. In this work, we drop the latter condition, replace it by a condition on the costs that discourages instability and recover the same result.

This class of problems is motivated by the following considerations. Controlled Markov chains are a popular paradigm for dynamic decision-making under uncertainty. Important application areas include control of queueing networks used to model computer and communication networks, flexible manufacturing systems etc., not to mention economic applications. In many of these situations, the problem calls for a simultaneous consideration of more than one optimization criterion. There are several ways to approach such multiobjective problems. One is to reduce it to a single objective problem by coalescing the objectives into one via a 'utility function'. However, the choice of such a function is not always obvious nor are the desirable ones necessarily the most amenable to analysis. An alternative approach, in tune with standard engineering practice, is to optimize one objective function (presumably the most important) while keeping the rest within reasonable bounds. This leads to the constrained control problem we discuss here.

A remark on the cost structure: In situations where money or effort contributed now is more expensive than that contributed later (due to interest, inflation etc.), one 'discounts' the future leading to the 'discounted cost control problems'. This is typical in economic applications. There are, however, situations where one plans for and anticipates a near-equilibrium (or 'steady state') behaviour for long stretches of time with no reason to favour short-term optimality over long-term optimality and with set-up costs, transients etc. either negligible or separately taken care of. The prime example of such is the case of computer and communication networks. Then it makes sense to consider the 'long-run average' or 'ergodic' cost as we do here whereby one averages the 'running cost' function with respect to the equilibrium distribution.

As for the assumptions used in Borkar (1991) and here, the former assumes a blanket stability, thereby ruling out a priori any possibility that the system may become unstable (e.g. the queue size blows up). This is often unreasonable in the applications mentioned above. The present set of assumptions allows instability but imposes a cost structure which penalizes instability and thereby ensures that the optimal system will also be stable.

The next section gives a precise statement of the problem. The main result is proved in § 4 following some preliminaries in § 3.

2. Notation and problem description

Let $X_n, n \geqslant 0$, be a controlled Markov chain on state space $S = \{1, 2, \ldots,\}$ with transition matrix $P_u = [[p(i,j,u_i)]]$, $i,j \in S$, indexed by the control vector $u = [u_1, u_2, \ldots]$. Here $u_i \in D(i)$ for $i \in S$, $D(i)$ being a prescribed compact metric space. As argued in Borkar (1989, p. 642), we may assume that the $D(i)$'s are identical copies of a fixed compact metric space D. Let $L = D^\infty$. The maps $p(i,j,.)$ are assumed to be continuous.

For any Polish space Y, let $P(Y) =$ the space of probability measures on Y with Prohorov topology (Billingsley 1968). If Y is countable, say $\{1, 2, \ldots\}$, write $\nu \in P(Y)$ as a row vector $[\nu(\{1\}), \nu(\{2\}), \ldots]$ or simply $[\nu(1), \nu(2), \ldots]$.

A control strategy is a sequence $\{Z_n\}$, $Z_n = [Z_n(1), Z_n(2), \ldots]$ of L-valued random variables such that for $i \in S, n \geqslant 1$,

$$P(X_{n+1} = i / X_m, Z_m, m \leqslant n) = p(X_n, i, Z_n(X_n)). \tag{1}$$

If $\{Z_n\}$ are independent and identically distributed (i.i.d.) with a common law $\Phi \in P(L)$, call it a stationary randomized strategy (SRS for short), denoted $\gamma[\Phi]$. As argued in (Borkar 1989, p. 642), we may take Φ to be a product measure $\Phi = \Pi \hat{\Phi}_i$ for $\hat{\Phi}_i \in P(D)$, $i \in S$. Under $\gamma[\Phi]$, $\{X_n\}$ is a Markov chain with a stationary transition matrix

$$P[\Phi] = \left[\left[\left[\int p(i,j,u) \hat{\Phi}_i(\mathrm{d}u) \right]\right]\right].$$

If Φ is a Dirac measure at some $\xi \in L$, call it a stationary strategy (SS for short), denoted $\gamma\{\xi\}$. The corresponding transition matrix is $P\{\xi\} = P_\xi$.

We assume throughout that S is a single communicating class under all SRS. Say that an SRS $\gamma[\Phi]$ (respectively an SS $\gamma\{\xi\}$) is a stable SRS or SSRS (respectively a stable SS or SSS) if the corresponding Markov chain is positive recurrent and thus has a unique invariant probability measure $\pi[\Phi]$ (respectively $\pi\{\xi\}$). Define the corres-

ponding 'ergodic occupation measure' $\hat{\pi}[\Phi] \in P(S \times D)$ by

$$\int f \, d\hat{\pi}[\Phi] = \sum \pi[\Phi](i) \int f(i,u) \hat{\Phi}_i(du), \, f \in C_b(S \times D).$$

($\hat{\pi}\{\xi\}$ is defined analogously.) Let $G = \{\hat{\pi}[\Phi] | \gamma[\Phi] \text{ an SSRS}\}$.

Let k_i: $S \times D \to R^+, 0 \leqslant i \leqslant m$, be continuous and $\alpha_i > 0, 1 \leqslant i \leqslant m$, be prescribed. Let $H \subset G$ be the set on which

$$\int k_i \, d\hat{\pi}[\Phi] \leqslant \alpha_i, \quad 1 \leqslant i \leqslant m. \tag{2}$$

We assume that H is nonempty and furthermore requires $\{k_i\}$ to satisfy

$$\liminf_{j \to \infty} \inf_u k_i(j,u) > \alpha_i, \quad 0 \leqslant i \leqslant m, \tag{3}$$

where

$$\alpha_0 = \inf_H \int k_0 \, d\hat{\pi}[\Phi],$$

is assumed to be finite. (3) is satisfied in particular for $\{k_i\}$ of the form $k_i(j,u) = f_i(j)$ where $f_i(j)$ increases to ∞ as $j \to \infty$.

We also make the following assumption, called 'stability under local perturbation' in Borkar (1989): If $\gamma[\Phi]$ is an SSRS and product measures $\Phi', \Phi \in P(L)$ differ in at most finitely many components, then $\gamma[\Phi']$ is an SSRS. As argued in Borkar (1989, p. 645), this is satisfied in particular when each $i \in S$ has only finitely many neighbours, i.e., $p(i,j,.) = 0$ for all but a finitely many j.

Our problem is to minimize over H the cost

$$\int k_0 \, d\hat{\pi}[\Phi]. \tag{4}$$

3. Preliminaries

This section establishes some technical lemmas, the first being recalled from Borkar (1989). Let $P_0 \subset L$ be the set of product probability measures on L.

Lemma 3·1. (*Borkar 1989*) *G is closed convex and its extreme points correspond to* SSS.

Proof. Let $\gamma[\Phi_1], \gamma[\Phi_2]$ be two SSRS with $\Phi_j = \Pi\hat{\Phi}_{ji}, j = 1, 2$. Let $0 \leqslant a \leqslant 1$ and define $\Phi = \Pi\hat{\Phi}_i \in P_0(L)$ by

$$\hat{\Phi}_i = (a\pi[\Phi_1](i)\hat{\Phi}_{1i} + (1-a)\pi[\Phi_2](i)\hat{\Phi}_{2i})/(a\pi[\Phi_1](i) + \\ + (1-a)\pi[\Phi_2](i)), \quad i \in S.$$

From this definition and the fact that

$$\pi[\Phi_i] \, P[\Phi_i] = \pi[\Phi_i], \quad i = 1, 2, \tag{5}$$

it is easily seen that

$$(a\pi[\Phi_1] + (1-a)\pi[\Phi_2])P[\Phi] = (a\pi[\Phi_1] + (1-a)\pi[\Phi_2]).$$

Thus, $\pi[\Phi] = a\pi[\Phi_1] + (1-a)\pi[\Phi_2]$. Then a straightforward computation leads to

$$\hat{\pi}[\Phi] = a\hat{\pi}[\Phi_1] + (1-a)\hat{\pi}[\Phi_2].$$

Convexity follows. Now let $\gamma[\Phi_n], n = 1, 2, \ldots$, be a sequence of SSRS such that $\hat{\pi}[\Phi_n] \to \nu$ for some $\nu \in P(S \times D)$. Let $\pi \in P(S)$ be the image of ν under the projection $S \times D \to S$. Then $\pi[\Phi_n] \to \pi$ in $P(S)$. Disintegrate ν(Schwartz 1961) as $\nu(\{i\} \times B) = \pi(\{i\})\hat{\Phi}_i(B)$ for $i \in S$ and B a Borel subset of D, with the regular conditional law $\hat{\Phi}_i \in P(D)$ for each i. Let $\Phi = \Pi\hat{\Phi}_i$. Since $p(.,j,.)$, $j \in S$, are continuous,

$$\int p(.,j,.)\,\mathrm{d}\hat{\pi}[\Phi_n] \to \int p(.,j,.)\,\mathrm{d}\nu, \quad j \in S.$$

Also, $\pi[\Phi_n] \to \pi$ in $P(S)$ implies by Scheffe's theorem (Billingsley 1968) that

$$\pi[\Phi_n] \to \pi \text{ in total variation.} \tag{6}$$

Putting the two together

$$\pi[\Phi_n]P[\Phi_n] \to \pi P[\Phi], \tag{7}$$

termwise. From (5)–(7), we have $\pi P[\Phi] = \pi$, i.e., $\pi = \pi[\Phi]$. Hence $\nu = \hat{\pi}[\Phi]$ and G is closed. Next, let $\gamma[\Phi], \Phi = \Pi\hat{\Phi}_i$, be an SSRS such that for some $i_0 \in S$ and $0 < a < 1$, there exist ϕ_1, ϕ_2 in $P(D)$ such that

$$\int p(i_0,.,u)\hat{\Phi}_{i0}(\mathrm{d}u) = a\int p(i_0,.,u)\phi_1(\mathrm{d}u) + (1-a)\int p(i_0,.,u)\,\phi_2(\mathrm{d}u), \tag{8}$$

$$\int p(i_0,.,u)\,\phi_1(\mathrm{d}u) \neq \int p(i_0,.,u)\,\phi_2(\mathrm{d}u), \tag{9}$$

as vectors, the integrations being termwise. Without any loss of generality, let $i_0 = 1$. Define $\Phi_i \in P_0(L)$ by $\Phi_i = \phi_i \times \Pi_{j=2}^{\infty}\hat{\Phi}_j, i = 1, 2$.

From the assumption of stability under local perturbations, it follows that $\gamma[\Phi_i]$, $i = 1, 2$, are SSRS. If $\pi[\Phi] = \pi[\Phi_1] = \pi[\Phi_2]$, the equation

$$\sum \pi[\Phi_i](k)\int p(k,j,u)\hat{\Phi}_{ik}(\mathrm{d}u) = \pi[\Phi_i](j), \quad i = 1, 2,$$

contradicts (9) for some j. Hence any two of $\pi[\Phi], \pi[\Phi_1], \pi[\Phi_2]$ are distinct from each other. Let $b \in (0, 1)$ be such that

$$a = b\pi[\Phi_1](1)/(b\pi[\Phi_1](1) + (1-b)\pi[\Phi_2](1)).$$

Argue as above to conclude that

$$\hat{\pi}[\Phi] = b\hat{\pi}[\Phi_1] + (1-b)\hat{\pi}[\Phi_2].$$

Thus $\hat{\pi}[\Phi]$ cannot be an extreme point of G, proving the second claim. QED

COROLLARY 3.1

H is closed convex.

Proof. Convexity follows easily from the above. Closedness follows on observing that (2) is preserved under sequential limits in G. QED

Lemma 3.2. Any element of H is the barycentre of a probability measure supported on the extreme points of H.

Proof. Let $\bar{S} = SU\{\infty\}$ denote the one point compactification of S. View $P(S \times D)$ as a subset of $P(\bar{S} \times D)$ by identifying each $\mu \in P(S \times D)$ with the unique $\bar{\mu} \in P(\bar{S} \times D)$ that restricts to μ on $S \times D$. Let $\bar{H}(\bar{G})$ be the closure of $H(G)$ in $P(\bar{S} \times D)$. Then $\bar{H}$ is compact. Let $H_e, \bar{H}_e$ be the sets of extreme points of $H, \bar{H}$ respectively. Suppose $\nu \in H_e \backslash \bar{H}_e$. Then ν is a convex combination of two distinct elements of $\bar{H}$ at least one of which must lie in $\bar{H} \backslash H$ and thus assign a strictly positive mass to $\{\infty\} \times D$. This is possible only if $\nu(\{\infty\} \times D) > 0$ which is false. Thus $H_e \subset \bar{H}_e$. By Choquet's theorem (Phelps 1966) each $\mu \in H$ is the barycentre of a probability measure η on $\bar{H}_e$. If $\eta(\bar{H}_e \backslash H_e) > 0$, then we must have $\mu(\{\infty\} \times D) > 0$, a contradiction. Thus $\eta(H_e) = 1$ and the claim follows. QED

Lemma 3.3. Each $\nu \in \bar{H}$ is of the form

$$\nu(A) = \delta \nu'(A \cap (S \times D)) + (1 - \delta)\nu''(A \cup (\{\infty\} \times D)) \tag{10}$$

for A Borel in $\bar{S} \times D$ where $\delta \in [0, 1]$, $\nu' \in G$ and $\nu'' \in P(\{\infty\} \times D)$.

Proof. That (10) holds for some $\nu' \in P(S \times D)$ is clear. Without loss of generality, let $\delta > 0$. If $\delta = 1, \nu \in H$ and the claim is immediate. If not, there exist $\{\hat{\pi}[\Phi_n]\}$ in H such that

$$\hat{\pi}[\Phi_n] \to \nu \text{ in } P(\bar{S} \times D).$$

For $j \in S$,

$$\int p(.,j,.)\, \mathrm{d}\hat{\pi}[\Phi_n] = \hat{\pi}[\Phi_n](\{j\} \times D).$$

Since $\{j\} \times D$ is both open and closed in $\bar{S} \times D$,

$$\hat{\pi}[\Phi_n](\{j\} \times D) \to \nu(\{j\} \times D) = \delta \nu'(\{j\} \times D). \tag{11}$$

Disintegrate ν' as

$$\nu'(\{i\} \times \mathrm{d}u) = \pi(\{i\})\phi_i(\mathrm{d}u)$$

for $\pi \in P(S), i \to \phi_i \colon S \to P(D)$. Set $\Phi = \Pi_i \phi_i$. For $N \geqslant 1$,

$$\sum_{i=1}^{N} \pi[\Phi_n](i) \int p(i,j,u)\hat{\Phi}_{ni}(\mathrm{d}u) \to \delta \sum_{i=1}^{N} \pi(\{i\}) \int p(i,j,u)\phi_i(\mathrm{d}u).$$

Thus

$$\liminf_{n \to \infty} \int p(.,j,.)\, \mathrm{d}\hat{\pi}[\Phi_n] \geqslant \delta \int p(.,j,.)\, \mathrm{d}\nu'. \tag{12}$$

Equations (11), (12) together imply

$$\int p(.,j,.)\,\mathrm{d}\nu' \leqslant \nu'(\{j\} \times D), \quad j \in S.$$

Both sides sum up to 1 when summed over j. Thus the equality must hold for each j. Thus

$$\sum_i \pi(\{i\}) \int p(i,j,u)\phi_i(\mathrm{d}u) = \pi(\{j\}), \quad j \in S.$$

Thus $\nu' = \hat{\pi}[\Phi]$, completing the proof. QED

Theorem 3.1 *Expression* (4) *attains its minimum at an extreme point of* H.

Proof. Let $\hat{\pi}[\Phi_n]$, $n \geqslant 1$, be a sequence in H such that

$$\int k_0\,\mathrm{d}\hat{\pi}[\Phi_n] \downarrow \alpha_0.$$

In view of the foregoing, we may drop to a subsequence if necessary and suppose that $\hat{\pi}[\Phi_n] \to \nu$ in $\bar{H}$ for a ν as in (10) with $\nu' = \hat{\pi}[\Phi]$ for some $\gamma[\Phi]$. Fix $j, 1 \leqslant j \leqslant m$. Pick $\varepsilon_j > 0, m_j \leqslant 1$ such that

$$\inf_u k_j(i,u) \geqslant \alpha_j + \varepsilon_j, \text{ for } i \geqslant m_j.$$

For $n \geqslant 1$, set

$$k_{jn}(i,u) = k_j(i,u)I\{i \leqslant m_j + n\} + (\alpha_j + \varepsilon_j)I\{i > m_j + n\}.$$

Then

$$\begin{aligned}\alpha_j &\geqslant \liminf_{l \to \infty} \int k_j\,\mathrm{d}\hat{\pi}[\Phi_l] \\ &\geqslant \lim_{l \to \infty} \int k_{jn}\,\mathrm{d}\hat{\pi}[\Phi_l] \\ &= \delta \int k_{jn}\,\mathrm{d}\hat{\pi}[\Phi] + (1-\delta)(\alpha_j + \varepsilon_j).\end{aligned}$$

Letting $n \to \infty$ on the right,

$$\alpha_j \geqslant \delta \int k_j\,\mathrm{d}\hat{\pi}[\Phi] + (1-\delta)(\alpha_j + \varepsilon_j).$$

This is possible only if $\delta > 0$ and

$$\int k_j\,\mathrm{d}\hat{\pi}[\Phi] \leqslant \alpha_j.$$

Since $j, 1 \leqslant j \leqslant m$, was arbitary, $\hat{\pi}[\Phi] \in H$. Now let $\varepsilon > 0$ and $m_0 \geqslant 1$ be such that

$$\inf_u k(i,u) \geqslant \alpha_0 + \varepsilon, \text{ for } i \geqslant m_0.$$

Argue as above to conclude

$$\alpha_0 \geqslant \delta \int k_0 \, d\hat{\pi}[\Phi] + (1-\delta)(\alpha_0 + \varepsilon)$$

and therefore

$$\int k_0 \, d\hat{\pi}[\Phi] \leqslant \alpha_0.$$

From the definition of α_0, equality holds. Thus (4) attains its minimum on H at $\hat{\pi}[\Phi]$. By lemma 3·2, $\hat{\pi}[\Phi]$ is the barycentre of a probability measure η on H_e. Thus

$$\alpha_0 = \int d\eta(\rho) \left(\int k_0 \, d\rho \right).$$

Since $\int k_0 \, d\rho \geqslant \alpha_0$ for $\rho \in H_e$, we must have $\int k_0 \, d\rho = \alpha_0$ for η-a.s. ρ, proving the claim. QED

4. Proof of the main results

Our main result is the following:

Theorem 4.1 *The constrained problem above has an optimal SSRS* $\gamma[\Phi], \Phi = \Pi\hat{\Phi}_i$, *satisfying: For all but at most m values of i,* $\hat{\Phi}_i$ *is a Dirac measure. For the remaining i, it is a convex combination of* $r_i \geqslant 2$ *Dirac measures where the integers* $\{r_i\}$ *satisfy* $\Sigma_i r_i \leqslant m$.

The proof proceeds in several steps. It helps at times to view G, H as subsets of the topological vector space of signed measures on $S \times D$. Let $m = 1$ for the time being. Let $\hat{\pi}[\Phi] \in H_e$ be such that $\int k_0 \, d\hat{\pi}[\Phi] = \alpha_0$. If $\hat{\pi}[\Phi]$ is an extreme point of G, $\hat{\pi}[\Phi] = \hat{\pi}\{\xi\}$ for some $\xi \in L$ by lemma 3.1 and we are done. Suppose not. Then it is a convex combination of two distinct elements ν_1, ν_2 of G. (Note that $\int k_1 \, d\nu_i < \infty$ for each i because $\int k_1 \, d\hat{\pi}[\Phi] < \infty$). If $\nu_1, \nu_2 \in H$, $\hat{\pi}[\Phi]$ cannot be in H_e. If $\nu_1, \nu_2 \in G \backslash H$, $\hat{\pi}[\Phi]$ cannot be in H because $G \backslash H$ is convex (recall $m = 1$). Thus exactly one of the ν_i's, say ν_2, is in $G \backslash H$. Then $\int k_1 \, d\nu_2 > \alpha_1$ and therefore $\int k_1 \, d\nu_1 < \alpha_1$. The value of $\int k_1 \, d\nu$ increases continuously from one strictly less than α_1 to one strictly exceeding α_1 as ν moves from ν_1 to ν_2 along the line segment joining the two. This segment must intersect the hyperplane $\{\nu | \int k_1 \, d\nu = \alpha_1\}$ at $\hat{\pi}[\Phi]$ because $\hat{\pi}[\Phi] \notin H_e$ otherwise. Now one can use a result of Dubins (1962) (see also Witsenhausen 1980, p. 265) to conclude that $\hat{\pi}[\Phi]$ is a convex combination of two distinct extreme points of $\bar{G}$. (Recall $H_e \subset \bar{H}_e$). If either of the latter assigns a strictly positive mass to $\{\infty\} \times D$, so would $\hat{\pi}[\Phi]$ itself. Thus both must be extreme points of G itself and hence are of the form $\hat{\pi}\{\xi_i\}$ for some $\xi_i \in L, i = 1, 2$. Let Z denote the line segment joining the two inclusive of the end points. Then $Z \subset P(S \times D)$ is compact.

The rest of the proof mimics Borkar (1991) closely with a few essential differences.

Lemma 4.1. *If* $\hat{\pi}[\Phi]$ *is a convex combination of* $\hat{\pi}[\Phi_i]$, $i = 1, 2$, *in G, the latter must lie on Z.*

Proof. If not, the intersection of the rectangle formed by $\hat{\pi}[\Phi_i], \hat{\pi}\{\xi_i\}, i = 1, 2$, with the hyperplane $\{\nu | \int k_1 \, d\nu = \alpha_1\}$ is a line segment in H containing $\hat{\pi}[\Phi]$ in its relative interior. This contradicts the fact $\hat{\pi}[\Phi] \in H_e$, proving the claim. QED

For $i = 1, 2$, let $\xi_i = [\xi_i(1), \xi_i(2), \ldots]$ and define $\xi'_n, n \geqslant 0$ by $\xi'_0 = \xi_1$, $\xi'_n = [\xi_2(1), \ldots, \xi_2(n), \xi_1(n+1), \xi_1(n+2), \ldots]$ for $n \geqslant 1$.

Lemma 4.2. $\hat{\pi}\{\xi'_n\} \in Z$ *for* $n \geqslant 0$.

Proof. We shall prove a more general result: For $\xi \in L$ such that $\xi(i) =$ either $\xi_1(i)$ or $\xi_2(i)$ for each i, $\gamma\{\xi\}$ is an SSS with $\hat{\pi}\{\xi\} \in Z$. Suppose that $p(1, ., \xi_1(1)) \neq p(1, ., \xi_2(1))$. [If not, take the least j for which $p(j, ., \xi_1(j)) \neq p(j, ., \xi_2(j))$.] Let ϕ_i denote the Dirac measures at $\xi_i(1)$, $i = 1, 2$. From the proof of lemma 3.1, it is clear that for $\Phi_i = \phi_i \times \Pi_{j=2}^{\infty} \hat{\Phi}_j, i = 1, 2, \hat{\pi}[\Phi_i], i = 1, 2$ are distinct and $\hat{\pi}[\Phi]$ is distinct from either and is a convex combination of the two. (By our assumption of stability under local perturbation, $\gamma[\Phi_i], i = 1, 2$, are SSRS.) By lemma 4.1, $\hat{\pi}[\Phi_i], i = 1, 2$, lie on Z. Repeat the argument with Φ_1 or Φ_2 in place of Φ and state 2 in place of state 1. It follows that for $\Phi' \in P(L)$ of the type: $\Phi' = \psi_1 \times \psi_2 \times \Pi_{j=3}^{\infty} \hat{\Phi}_j$, where ψ_i are Dirac measures at either $\xi_1(i)$ for $\xi_2(i)$ for $i = 1, 2, \hat{\pi}[\Phi'] \in Z$. Iterating the argument, we have: for any $\Phi' = \Pi \hat{\Phi}_i$, such that for some $n \geqslant 1, \hat{\Phi}'_i = \hat{\Phi}_i$ for $i > n$ and the Dirac measure at either $\xi_1(i)$ or $\xi_2(i)$ for $i \leqslant n$, we have $\hat{\pi}[\Phi'] \in Z$. Let Q denote the set of such Φ'. Then there exists a sequence $\{\Phi'_n\}$ in Q such that $\Phi'_n \to$ the Dirac measure at ξ as $n \to \infty$. Since Z is tight, we may drop to a subsequence if necessary and suppose that $\pi[\Phi'_n] \to \pi$ in $P(S)$ (and hence in total variation) for some π. Letting $n \to \infty$ in

$$\int p(., j, .) \, d\hat{\pi}[\Phi'_n] = \hat{\pi}[\Phi'_n](\{j\} \times D), \quad j \in S,$$

we have

$$\sum \pi(i) p(i, j, \xi(i)) = \pi(j), \quad j \in S.$$

Thus $\pi = \pi\{\xi\}$. It is now easy to check that $\hat{\pi}[\Phi'_n] \to \hat{\pi}\{\xi\}$. Since Z is closed, the claim follows. QED

Consider Z as a union of two closed line segments Z_1 and Z_2, Z_1 (respectively Z_2) being the line segment joining $\hat{\pi}\{\xi_1\}$ (respectively $\hat{\pi}\{\xi_2\}$) with $\hat{\pi}[\Phi]$. By lemma 4.2, $\hat{\pi}\{\xi'_0\} \in Z_1$ and as $n \to \infty$, $\hat{\pi}\{\xi'_n\}$ moves from Z_1 to Z_2. Then either $\hat{\pi}\{\xi'_n\} = \hat{\pi}[\Phi]$ for some n (and we are done) or there is an n such that $\hat{\pi}\{\xi'_{n-1}\} \in Z_1, \hat{\pi}\{\xi'_n\} \in Z_2$. But ξ'_{n-1}, ξ'_n differ only in their nth component and $\hat{\pi}[\Phi]$ is a convex combination of the two. Argue as in lemma 3.1 to conclude that $\hat{\pi}[\Phi] = \hat{\pi}[\hat{\Phi}]$ for $\hat{\Phi} = \Pi \hat{\Phi}_j$ satisfying: $\hat{\Phi}_i$ is a Dirac measure at $\xi_1(i)$ (respectively $\xi_2(i)$) for $i > n$ (respectively $i < n$) and a convex combination of Dirac measures at $\xi_1(n), \xi_2(n)$ for $i = n$. This proves theorem 4.1 for $m = 1$. The general case follows by induction, replacing G at the nth induction step ($n < m$) by $G \cap \{\hat{\pi}[\Phi] | \int k_j \, d\hat{\pi}[\Phi] \leqslant \alpha_j, 1 \leqslant j \leqslant n\}$.

The following then follows immediately from standard Lagrange multiplier theory (Luenberger 1967, pp. 216–219).

Theorem 4.2 *Suppose that* (2) *is a strict inequality for all* $j, 1 \leqslant j \leqslant m$, *and some* $\hat{\pi}[\Phi'] \in H$. *Then there exist numbers* $\lambda_1, \ldots, \lambda_m \in R^+$ *such that the map*

$$\nu \to \int k_0 \, d\nu + \sum_{i=1}^{m} \lambda_i \left(\alpha_i - \int k_i \, d\nu \right)$$

attains its minimum on G at the $\hat{\pi}[\Phi]$ *of theorem 4.1. Moreover, the following holds for* $v \in G$ *and* $\mu_1, \ldots, \mu_m \in R^+$:

$$\int k_0 \, dv + \sum_{i=1}^{m} \lambda_i \left(\alpha_i - \int k_i \, dv \right) \geqslant \int k_0 \, d\hat{\pi}[\Phi] + \sum_{i=1}^{m} \lambda_i \left(\alpha_i - \int k_i \, d\hat{\pi}[\Phi] \right)$$

$$\geqslant \int k_0 \, d\hat{\pi}[\Phi] + \sum_{i=1}^{m} \mu_i \left(\alpha_i - \int k_i \, d\hat{\pi}[\Phi] \right).$$

References

Altman E, Shwartz A 1990 Sensitivity of constrained Markov decision processes, EE Pub. No. 741, Dept. of Electrical Eng., Technion, Haifa, Israel

Beutler F J, Ross K W 1985 Optimal policies for controlled Markov chains with a constraint, *J. Math. Anal. Appl.* 112: 236–252

Billingsley P 1968 *Convergence of probability measures* (New York: Wiley)

Borkar V S 1989 Control of Markov chains with long-run average cost criterion: the dynamic programmin equations. *SIAM J. Control Optim.* 27: 642–657

Borkar V S 1991 Topics in controlled Markov chains, *Pitman research notes in mathematics* (Harlow: Longman) Chap. 7

Dubins L 1962 On extreme points of convex sets, *J. Math. Anal. Appl.* 5: 237–244

Hordijk A, Kallenberg L C M 1984 Constrained undiscounted stochastic dynamic programming. *Math. Oper. Res.* 9: 276–289

Luenberger D 1967 *Optimization by vector space methods* (New York: Wiley)

Phelps R 1966 *Lectures on Choquet's theorem* (New York: Van Nostrand)

Ross K W 1989 Randomized and past-dependent policies for Markov decision processes with multiple constraints. *Oper. Res.* 37: 474–477

Schwartz L 1961 *Disintegration of measures* (Bombay: Tata Institute of Fundamental Research)

Witsenhausen 1980 Some aspects of convexity useful in information theory. *IEEE Trans. Inf. Theory* IT-26: 265–271

Author Index

Ackley D H
see Hinton G E 1
Agrawala A 133, 137
Albin S L 59
Aldous D J 120
Altman E 155
Ancker C Jr 56
Arnold L 70
Astom K J 147, 152

Baccelli F 132, 139, 140, 142
Balbo G 82
see Marsan M A 94
Bambos N 117
Banerjee K 10
Banerjee S 11, 29
see Sastry P S 10, 29
Barlow R E 140
Basharin G 59
Baskett F 58, 108, 109
Becker A 147
Becker S T 110
Bertsekas D 115, 119, 120, 121, 124
Beuerman S L 122
Beutler F J 155
Bhat U N 69
Billingsley P 9, 156, 158
Bitran G R 56, 57, 58, 59
Blake A 1, 12
Bonomi F 138
Borkar V S 155, 156, 157, 161
Boxma O J 62
Brill P H 108
Bruell S C
see Balbo G 82
Bush R R 13, 17
Bux W 124
Buzacott J A 56, 58, 81, 94
Buzen J P 109

Caines P E
see Goodwin G C 147
Capetanakis J I 126
Chandy K M 81, 84, 109
see Baskett F 58, 108, 109
Chang X 126
Chen H F 148, 152, 154
Cheng X 62
Chillingworth D R J 78
Chow Y C 138
Cidon I 127
Clare L P 121
Claude Fort J
see Cottrell M 119
Conte G
see Marsan M A 94
Conway R W 108
Cottrell M 119
Coyle E J 122
Cunningham G A 123

Davis L 1
Dekker H 71
de Moraes L F M 128
de Souza e Silva E 138
Dobson G 55
Dong P
see Geman D 11
Dubins L 161

Eager D L 134
Eisenberg M 124, 125
Ephremides A 118, 133, 134, 135, 137

Fabio R
see Mitra D 4
Flatto L 137, 139
Flores C 137
Foschini G J 136
Fuhrmann S W 61, 62

Gafarian A V 56
Gallagher R 115, 119, 120, 121, 124
Gantmacher R 38
Gascon A 55
Gaver D P 70
Gelatt C D
see Kirkpatrick S 4
Geman D 10, 11
Geman S 10, 11
see Geman D 11
Georgiadis L
see Paterakis M 116
Georgiopoulos M 128
Gerla M 138
Gershwin S B 55
Ghanta S
see Balbo G 82
Ghez S 121
Gocmen M 57
Goodman J 119, 120, 121
Goodwin G C 147
Gordon W J 108
Graffigne C 11
see Geman S 11
see Goodman J 119
Greenberg A G
see Goodman J 120, 121
Gross D 100, 102
Gu X D 29
Gun L 141, 142, 148, 152, 154

Hahn S 139
Hajek B 119, 120, 123, 136, 137, 138, 143
Haken H 79
Hanumantha Rao P 81, 92
Hasegawa T
see Takine T 62
Hashida O 124, 125
Hastad J 120, 121
Herzag
see Chandy K M 81, 84, 109
Hinton G E 1
Hirsch M W 34, 35, 38, 48, 49
Ho Y C 81, 93
Hopfield J J 2, 33, 34, 39, 40, 41, 43, 49
Hordijk A 155
Humblet P A 128
Hummel R A 1, 5
Hunt V B 121
Hwang K 138

Ibe O C 62, 126
Inan K M 94

Jackson J R 108
Jacquet P 127
Jaiswal M P 71, 72, 74, 76, 77
Jaiswal N K 69, 70, 71
Jean-Marie A 132, 141, 142

Kallenberg L C M 155
Karmarkar U S
see Dobson G 55

Karmeshu 69, 70, 71
see Jaiswal M P 71, 72, 74, 76, 77
Kazakos D 119, 120, 123
Kelly F P 58
Kemae T 134
Khasminskii R Z 128
Kirkpatrick S 4
Kleinrock L 53, 69, 78, 107, 108, 121, 122
Kobayashi H 69, 109
Kohler W 138
Konheim A G 124
Krengel U
see Kemae T 134
Krishna Prasad K R 93, 102
Kuehn P J 56, 126
Kumar A 131, 132, 138, 141, 142, 143
Kumar P R 55, 147, 148, 151
see Becker A 147
see Chen H F 148
Kuri J 132, 141, 142, 143
Kushner H 9

Lai T L 148, 151
Lakshmivarahan S 17, 19
Lavenberg S S 59, 69, 105, 109
Lazowska E D
see Eager D L 134
Lea C-T A 121, 122
Leachman R C 55
Lehoczky J P 70
Leighton T
see Hastad J 120, 121
Leuenberger D 162
Levy H 53
Lim J T 119, 120
Lin W 132
Liu B 122
Loynes R 116

Madras N
see Goodman J 119, 120, 121
Makowski A M 132
see Bacceli F 139, 142
Malgouyres G
see Cottrell M 119
Manjunath M S 102
March P 119
see Goodman J 119, 120, 121
Marie R A 56
Marroquin J 3
Mars P 29
Marsan M A 94
Mason L G 29
Massey J L 126
Massey W A
see Baccelli F 139, 140
Maxwell W L
see Conway R W 108
McClelland J L 1, 12
McKean H P 137
Medith J S 121, 122, 123
Meenakshisundaram C R 81, 82
Meerkov S M 119, 120
Mehravari N 128
Meister B W 62, 124
Merakos L 119, 120
see Georgiopoulos M 128
Mikhailov V A 118, 126, 127
Miller D R 100, 102
Miller L W
see Conway R W 108
Mirchandaney R 134
Mitra D 4
Mitter S
see Marroquin J 3
Moler C 102
Mosely J 128
Mosteller F 13, 17
Mukhopadhyay S 25
Muntz R R
see Baskett F 58, 108, 109
Muralidharan S 10
Murata T 82

Narahari Y 81, 93, 94, 102
Narendra K S 8, 13, 18, 20, 29, 30
Nelson R 69, 72, 73, 78, 79, 80, 119, 137, 139
Neuts M F 123
Newell G F 108
Ni L M 138
Nof S Y
see Seidmann A 62
Noguchi S 119

O'Brien G L
see Kemae T 134
Onozato Y 119
Ott T J 108

Pakes A G 122
Palacios F G
see Baskett F 58
Palacios-Gomez F
see Baskett F 108, 109
Papatani-Kazakos P
see Georgiopoulos M 128
see Paterakis M 116
Paterakis M 116
Perkins J R 55
Phelps R 159
Philips T K 137
Poggio T
see Marroquin J 3
Posner M J M 108
Poston T 79
Prathima B V 117, 118
Proschan F 140

Raith T 62
Ram R 51
Ramadge P J
see Goodwin G C 147
Ramakrishnan K R 22, 23
see Banerjee K 10
see Sastry P S 10, 29
Ramesh N R 82
Ranky P G 81
Rege K M 105, 110
see Becker S T 110
Reibman A 102
Reiman M I 136
Reiser M 109
Ren Wei 147
Rivest R L 120
Rogoff B
see Hastad J 120, 121
Rolski T 117
Rootzen H 147, 148
Rosenfeld A 1
Rosenkrantz W A 119, 120
Ross K W 155
Ross S M 100, 134
Robin I 119, 122
Rumelhart D E 1, 12
Rummel J
see Dobson G 55

Saadawi T N 118
Sachs S R 115, 121, 124, 128
Saksonov E A 124, 125
Salz J 136
Sastry P S 1, 8, 9, 10, 20, 25, 27, 29
see Banerjee K 10
Sauer C H 81
Schwartz A
see Baccelli F 139
Schwartz L 158
Schwartz M 136
Schwartz S C
see Ghez S 121
Schweitzer P J
see Seidmann A 62
Seidman A 62
Seidman T I 55
Sejnowsky T J
see Hinton G E 1
Sengupta B
see Becker S T 110
Servi L D 126
Shanthikumar J G 56, 57, 58
Shapiro I J 18

Sharma C L
see Jaiswal M P 71, 72, 74, 76, 77
Sharma V 115, 118, 119
Shorey R 139, 140, 141
Shwartz A 155
Sidi M 127
Smale S 38
Smiith D R 134
Solo V 148
Srinivasan M M 62
Stankovic J A
see Mirchandaney R 134
Sternby J 147, 148
Stewart I 79
Stoyan D 135
Sunder R 121
Suri R 94
Swartz G B 124
Szpankowski W 118, 127

Takagi H 53, 55, 61, 62, 124, 125
Takahashi Y
see Takine T 62
Takine T 62
Taneubaum A 142
Tank D W 2, 33, 34, 35, 39, 40, 41, 43
Tantawi A N 138, 139
Tapiero C S 70
Tedijanto 55, 62
Thathachar M A L 8, 9, 13, 19, 20, 25, 27, 29, 30
Thomopoulous S C A 121, 123
Tirupati D 56, 57, 58, 59
Tobagi F A 121, 122
Towsley D F 109, 119, 120, 138
see Baccelli F 139, 140
see Mirchandaney R 134
Tran-Gia P 62
Tripathi S 133, 137
Trivedi K S 95, 96, 102
Tsetlin M L 13
Tsitsklis J N 120
Tsybakov B S 118, 126, 127

van Kampen N G 71
Van Loan C F 102
van Schuppen J H
see Chen H F 148
Varaiya P
see Ephremides A 94, 133, 135, 137
Varma S 141, 142
Vacchi M P
see Kirkpatrick S 4
Venkatesh Y V
see Banerjee K 10
Verdu S
see Ghez S 121
Vidyasagar M 33, 42
Vincentelli A S
see Mitra D 4
Viswanadham N 51, 81, 93, 94, 102

Walrand J 117, 136, 137
see Ephremides A 133, 135, 137
Waltz L D 1
Wang Y T 61, 62
Weber R R 133, 134
Wei C Z 148, 151
see Becker A 147
Whitt W 56, 57, 58, 59, 134, 135
Williams R J 30
Winston W 134, 135
Witsenhausen 161
Wittenmark B 147, 152
Wolff R M 107
Wolff R W 65, 133, 134, 135, 137, 143
Wonham W M 94
Woo L
see Chandy K M 81, 84, 109

Yao D D 81, 94
Yashkov S F 108
Yum T S 133, 135

Zahorjan J
see Eager D L 134
Zhdanov V S 124, 125
Zisserman A 1, 12
Zucker S W 1, 5

Subject Index

Adaptive control
  Recent results on least squares-based adaptive control of linear stochastic systems in white noise 147
Artificial neural networks
  Stochastic automata and learning systems 13
Automated manufacturing system
  Stochastic analysis of versatile workcentres 51
Automated manufacturing systems
  Markovian models for deadlock analysis in automated manufacturing systems 93

Channel access
  Stability of multiple access communication networks 115
Consistent labelling problem
  Stochastic networks for constraint satisfaction and optimization 1
Control of queues
  Task allocation in multiserver systems – A survey of results 131
Control under constraints
  Controlled Markov chains with constraints 155
Controlled Markov chains
  Controlled Markov chains with constraints 155

Diffusion approximation method
  Queueing models with threshold phenomena: With special reference to machine interference problems 69
Discrete event systems
  Markovian models for deadlock analysis in automated manufacturing systems 93

Generalised stochastic Petri nets
  Integrated analytical models for flexible manufacturing systems 81

Integrated analytical models
  Integrated analytical models for flexible manufacturing systems 81
Inventory level
  Stochastic analysis of versatile workcentres 51

Lead time
  Stochastic analysis of versatile workcentres 51
Learning automata
  Stochastic networks for constraint satisfaction and optimization 1
Least-squares estimates
  Recent results on least squares-based adaptive control of linear stochastic systems in white noise 147
Load balancing
  Task allocation in multiserver systems – A survey of results 131

Machine interference problems
  Queueing models with threshold phenomena: With special reference to machine interference problems 69
Machine utilisation
  Stochastic analysis of versatile workcentres 51
Markovian models
  Markovian models for deadlock analysis in automated manufacturing systems 93
Mean time to deadlock
  Markovian models for deadlock analysis in automated manufacturing systems 93
Multiple access schemes
  Stability of multiple access communication networks 115

Neural networks
  Stochastic networks for constraint satisfaction and optimization 1
Neural networks
  An analysis of the equilibria of neural networks with linear interconnections 33

Optimal strategy
  Controlled Markov chains with constraints 115

Performance analysis of computer systems
  Multi-class queueing models for performance analysis of computer systems 105
Product from queueing networks
  Integrated analytical models for flexible manufacturing systems 81

Queueing models
  Queueing models with threshold phenomena: With special reference to machine interference problems 69
Queueing networks
  Multi-class queueing models for performance analysis of computer systems 105

Recursive parameter estimation
Recent results on least sqaures-based adaptive control of linear stochastic systems in white noise 147
Routing
Task allocation in multiserver systems – A survey of results 131

Sigmoid nonlinearities
An analysis of equilibria of neural networks with linear interconnections 33
Stability
Stability of multiple access communication networks 115
Stability of equilibria
An analysis of the equilibria of neural networks with linear interconnections 33
Stationary strategy
Controlled Markov chains with constraints 155
Stochastic adaptive control
Recent results on least squares-based adaptive control of linear stochastic systems in white noise 147
Stochastic analysis
Stochastic analysis of versatile workcentres 51
Stochastic automata
Stochastic automata and learning systems 13
Stochastic behaviour
Queueing models with threshold phenomena: With special reference to machine interference problems 69

Threshold behaviour
Queueing models with threshold phenomena: With special reference to machine interference problems 69

ACADEMY PUBLICATIONS IN ENGINEERING SCIENCES

General Editor: R Narasimha

Volume 1. The Aryabhata Project (eds U R Rao, K Kasturirangan)
Volume 2. Computer Simulation (eds N Seshagiri, R Narasimha)
Volume 3. Rural Technology (ed. A K N Reddy)
Volume 4. Alloy Design (eds S Ranganathan, V S Arunachalam, R W Cahn)

...written by eminent scientists of India and abroad... Academy deserves all praise in bringing out these [papers]

Powder Metall. Assoc. India, Newslett.

Volume 5. Surveys in Fluid Mechanics (eds R Narasimha, S M Deshpande)

An informal and stimulating publication... (provides) a survey of many important topics in Fluid Mechanics... All the papers are of excellent technical content and most are very well written. Many include lengthy reference lists, which are as useful as the body of the paper... The publishing quality is very good...

IEEE J. Ocean Eng.

...The general level (of papers) is high... Several are likely to have wide appeal...

J. Fluid Mech.

Volume 6. Wood Heat for Cooking (eds K Krishna Prasad, P Verhaart)

...interesting and stimulating account of technical thinking on the wood stove problem up to date... excellent collection of valuable and thought-provoking investigations.

Rev. Projector (India)

Volume 7. Remote Sensing (eds B L Deekshatulu, Y S Rajan)

...Several contributions are specifically addressing topics of national interest, however, the majority of the papers is of general interest for a wide community. The book can serve not only as an inventory of the Indian activities, but also as a textbook on techniques and applications of remote sensing.

Photogrammetria

...particularly refreshing...

Int. J. Remote Sensing

Volume 8. Water Resources Systems Planning (eds M C Chaturvedi P Rogers)

...well got up and very well printed... forms a valuable addition to our limited literature on Water Resources of India.

Curr. Sci.

Volume 9. Reactions and Reaction Engineering (eds R A Mashelkar, R Kumar)

...eighteen portraits of some major frontiers of research in CRE by a Micro-Who's Who of contemporary CRE researchers... good mix of theory and experiment that will provide many a good read for the specialist as well as the generalist researcher... "must-buy" for all research libraries, corporate or personal that contain CRE titles.

Can. J. Chem. Eng.

Volume 10. Reliability and Fault-Tolerance Issues in Real-Time Systems (ed. N Viswanadham)

...should help the reader to obtain a global view of the design of real-time systems with specifications in reliability and fault tolerance.

IEEE Control Syst. Mag.

Volume 11 Composite Materials and Structures (ed. K A V Pandalai)

...will surely find its place in libraries dealing with applied research.

Bull. Electrochem.

Volume 12. Developments in Fluid Mechanics and Space Technology (eds R Narasimha, A P J Abdul Kalam)

Volume 13. Advanced Ceramics (ed. E C Subbarao)

...presents a comprehensive view of advanced ceramics....should be of interest to students, research scientists, development engineers and to those engaged in the modern ceramic industry.

Ind. Ceram.

...most of the chapters put stress not only on the science, but also on the technological aspects.

Trans. Indian Ceram. Soc.